논문 2010-47CI-1-16

자연어 기반 온톨로지 질의 모듈 연구

(A Study on the Ontology Query Module based on Natural Language)

김 원 필*, 공 현 장**

(Won-Pil Kim and Hyun-Jang Kong)

요 약

온톨로지 내의 효율적 정보 탐색을 위한 질의 처리 부분은 온톨로지 응용의 측면에서 반드시 해결되어야 할 부분이다. 기존의 온톨로지에 대한 질의 처리 시스템은 온톨로지에 표현된 사실만을 분석하여, 단순 구조적 사실 정보만을 사용자에게 제공함으로써 데이터베이스 시스템 및 텍스트 기반 정보처리 시스템과 크게 다른 점이 없었다. 사실상, 기존의 온톨로지 질의 시스템들에서 사용하고 있는 질의 언어의 구조나 형식이 데이터베이스 시스템의 질의 언어(SQL)에 모태를 두고 있으므로, 온톨로지와 데이터베이스의 질의 처리 및 그 결과는 거의 동일하다. 이에 본 연구에서는 온톨로지 사용의 효율성을 극대화하기 위해 온톨로지에 대한 단순 질의 처리가 아닌, 추론 규칙에 기반한 추론된 사실들을 모두 처리할 수 있는 온톨로지 추론 및 질의에 대한 통합 시스템 개발의 필요성을 인식하고, 온톨로지에 대한 효율적 질의 처리 방법을 연구하였다.

Abstract

For an application of ontology, query processing is mandatory field for efficient information search in the ontology. Other query processing systems tend to analyze only facts and to simply provide structural information for users. In fact, the systems do not have big difference with database systems or text based information processing systems. Therefore, in this research, the method which can provide the inferred information based on axioms is suggested in order to maximize reusability of ontology.

Keywords: Ontology, RQL, RDQL, SPARQL

I. 서 론

온톨로지에 대한 연구와 이를 응용하고자 하는 분야들 사이에는 실제 온톨로지 활용을 위해 해결해야 할 많은 문제들이 존재하고 있다. 특히, 온톨로지 내의 효율적 정보탐색을 위한 질의처리 부분은 온톨로지 응용의 측면에서 반드시 해결되어야 할 부분이다. 기존의 온톨로지에 대한 질의처리 시스템은 온톨로지에 표현된 사실만을 분석하여, 단순 구조적 사실 정보만을 사용자에게 제공함으로써 데이터베이스 시스템 및 텍스트 기반 정보처리 시스템과 크게 다를 점이 없었다. 따라서, 온톨로지 질의를 위해서 많은 연구가 진행되어져 왔으며, RQL, RDQL, SPARQL과 같은 질의 언어들이 개발되었다[1]. 이러한 언어들의 구조는 데이터베이스 질의 언어인 SQL에 모태를 두고 있다. RQL, RDQL, SPARQL은 RDF 그래프 모델에 기반하여 사용자 질의를 처리함으로써, OWL 파일을 처리하는데 한계를 가지고 있으며, 질의를 수행하기 위해 복잡한 질의 구조를 사용하여 질의를 작성해야 하는 어려움이 있다. 따라서 본 연구에서는 기존 온톨로지 질의 언어(RQL, RDQL, SPARQL)들이 가지고 있는 질의 구문 작성 시 복잡성을 해결하기 위해 유도 구문 기반의 질의 처리가 가능한 질의 처리 인터페이스를 설계 구축하였다.

* 정회원, 조선이공대학교 사이버보안과 교수
(Chosun college university of science & technology)
** 정회원, 조선대학교병원 전산실
(Chosun university hospital)
접수일자: 2009년9월25일, 수정완료일: 2010년1월8일

Ⅱ. 관련 연구

RQL, RDQL, SPARQL과 같은 질의 언어들의 구조는 SQL에 모태를 두고 있으며 SQL은 데이터베이스의 일반화로 질의 언어의 사용이 보편화 및 구조화되어 있지만, 역시 데이터베이스를 처리하는데 적합한 언어로써 온톨로지를 처리하는 데는 한계가 있다. RQL]은 RACER의 사용을 위해 확장된 쿼리언어로 모델이론 기술논리에 기반을 둔 언어이다[2].

RQL은 사용자들에게 복합 쿼리를 이용하여 다른 DL 시스템들에 비해 정보검색 좋은 성능을 제공하지만 스키마 쿼리 메커니즘을 제공하지 않고 있는 단점이 있다. RDQL은 그래프 패턴으로 이루어져 있고 그래프로부터 정보를 얻으며 관계형 데이터베이스의 SQL과 비슷하게 구현되었다[3].

RDQL은 RDF와 OWL에 대한 매우 유연한 온톨로지 쿼리 언어이다. 하지만 논리곱을 지원하지 않고 스키마 쿼리를 제공하지 않는 등의 문제점이 있다. SPARQL은 RDF 그래프로부터 정보를 추출하는 쿼리 온톨로지 언어이다[4]. 기존의 RDQL에 쿼리 결과에 부가적인 정보 추가, 그래프 패턴의 논리합, 강화된 표현력, 명시적 그래프, 정렬의 기능이 더해진 SPARQL은 여러 기술들이 복잡하게 섞여있는 현재의 시맨틱 웹의 상황에서 표준 온톨로지 쿼리 언어로 선택될 가능성이 가장 높은 질의 언어이다[5].

최근의 연구에서는 온톨로지를 데이터베이스화 함으로써 데이터베이스로 변형된 내용에 대하여 SQL을 사용하여 질의를 처리하는 연구도 있었다. 대표적인 데이터베이스 기반 온톨로지는 워드넷이며, 이를 처리하기 위해서 SQL 구문이 사용된다. 표 1은 기존 온톨로지 질의 언어의 질의 처리 방식의 특징을 보여주고 있다.

표 1. 온톨로지 질의 언어 비교
Table 1. Comparision of Ontology Query Language.

	RQL,RDQL, SPARQL	SQL
처리영역	RDF	Database
추론기능	없음	없음
기본구조	SELECT, FROM, WHERE	
특징	RDF 그래프 모델에 기반한 텍스트 매칭의 온톨로지내의 간단한 구조적 정보 검색	온톨로지 검색을 위해서는 온톨로지 파일을 데이터베이스화하는 사전 작업이 요구됨

Ⅲ. 제안한 질의 처리 모듈

본 연구에서는 기존의 Select, From, Where 질의 구조를 트리플 입력에 맞게 변형하여 본 연구의 질의 처리 모듈에 적용하였다. 본 시스템에서 사용되는 질의 처리 언어는 Step1:Subject 추출, Step2:트리플 추출, Step3:트리플집합 Sub_2 생성, Step4:Subject 찾기, Step5:조건 검색에 관한 정의와 같이 다섯 단계로 구성되어져 있다.

1. Step 1:Subject 추출

추론된 트리플에 기반한 질의 언어에서 우선 몇 가지의 OWL 어휘의 의미를 이해하고 있어야 한다. 이중에 가장 중요한 한 가지가 바로 rdf:type 이다. 본 연구의 질의 처리 모듈에 처음 입력되는 값에 해당되는 모든 내용들은 바로 rdf:type으로 연결된 값이다. 해당 모듈은 다음과 같다.

```
select {SUBJECT}
 from {TRIPLE_SET}
where {rdf:type=:ls_arg1}
```

여기에서는 추론 규칙에 의해서 만들어진 트리플 집합에서 입력된 값과 rdf:type의 관계를 이루고 있는 모든 값들이 바로 사용자가 원하는 Subject가 될 수 있다. 본 연구에서는 미리 추론과정을 거쳐서 만들어진 추론된 트리플 집합을 사용함으로써 위의 모듈에서 더 풍부한 값들을 기대할 수 있지만 일반적인 추론되지 않은 구조적 트리플에서도 위의 모듈을 통해서 원하는 Subject를 추출할 수 있다.

```
select {SUBJECT}
 from Wine.triple
where rdf:type=:wine
```

wine과 rdf:type의 관계를 이루고 있는 트리플내의 모든 Subject를 찾는 질의이다. 본 질의의 결과는 아래와 같으며, 결과값은 STEP 2의 입력으로 사용된다. 위 예제에 대한 결과 값은 다음과 같다.

ChateauChevalBlancStEmilion,ChateauDYchemSauterne,ChateauLafiteRothschildPauillac,ChateauMorgonBeaujolais,CorbansPrivateBinSauvignonBlanc,ElyseZinfandel,FormanCabernetSauvignon,GaryFarrellMerlot,StGenevieveTexasWhite,WhitehallLaneCabernetFranc,VentanaCheninBlanc,StonleighSavignonBlanc,SelaksSauvignonBlanc,SelaksIceWine,SeanThackreySiriusPetiteSyrah,SchlossVolradTrochenbierenausleseRiesling,SchlossRothermelTrochenbierenausleseRiesling,SaucelitoCanyonZinfandel1998,SaucelitoCanyonZinfandel,StGenevieveTexasWhite,SantaCruzMountainVineyardCabernetSauvignon,PulignyMontrachetWhiteBurgundy,PeterMccoyChardonnay,PageMillWineryCabernetSauvignon,MountadamRiesling,MountadamPinotNoir,MountadamChardonnay,MountEdenVineyardEstatePinotNoir,MariettaZinfandel,MariettaPetiteSyrah,MariettaOldVinesRed,MariettaCabernetSauvignon,LongridgeMerlot,LaneTannerPinotNoir,KathrynKennedyLateral,KalinCellarsSemillon,GaryFarrellMerlot,FoxenCheninBlanc,FormanChardonnay,FormanCabernetSauvignon,ElyseZinfandel,CotturiZinfandel,CortonMontrachetWhiteBurgundy,CorbansSauvignonBlanc,CorbansPrivateBinSauvignonBlanc,CorbansDryWhiteRiesling,CongressSpringsSemillon,ChiantiClassico,ChateauMargaux,ChateauLafiteRothschildPauillac,ChateauDeMeursaultMeursault,ChateauDYchemSauterne,ChateauChevalBlancStEmilion

2. Step 2:트리플 추출

STEP 1에서 추출된 개념들을 Subject로 한 트리플 집합 내의 모든 Sub 트리플들을 추출한다. STEP 3을 진행하기 위하여 새로운 트리플 집합을 STEP 1의 결과 값을 기반으로 구성한다. 이렇게 새롭게 트리플을 구성함으로써 질의 처리 속도를 줄일 수 있다. STEP 2를 거쳐 새로운 트리플 집합_sub1이 만들어진다.

3. Step 3:트리플 집합 Sub_2 생성

트리플 집합_sub1에서 입력된 Predicate와 관련된 모든 트리플들을 추출하여 트리플 집합_sub2를 만든다. STEP 3에서 arg2는 반드시 predicate 정보가 입력되므로 본 연구의 질의처리 모듈에서 트리플내의 질의 처리는 predicate를 조사하여 새로운 트리플 집합_sub2를 만든다. 해당 모듈은 다음과 같다.

```
select {TRIPLE}
 from {TRIPLE_SET_sub1}
where
 {TRIPLE_SET_sub1.predicate=:ls_arg2}
```

여기에서, 다시 트리플 집합_sub2가 만들어지며, 이렇게 만들어진 트리플 집합_sub2는 STEP 4의 입력이 된다.

```
select {PREDICATE}
 from Wine.triple_sub1
where
 Wine.triple_sub1.predicate=:has_color
```

STEP 2에서 만들어진 Wine.triple_sub1 트리플 집합에서 has_color를 가지고 있는 모든 트리플들을 선택하여, 해당 트리플을 Wine.triple_sub2로 새롭게 구성한다. 구성된 결과는 다음과 같다.

ChateauChevalBlancStEmilion has_color Red, ChateauDYchemSauterne has_color Red,
ChateauLafiteRothschildPauillac has_color Red, ChateauMorgonBeaujolais has_color Red,
CorbansPrivateBinSauvignonBlanc has_color Red, ElyseZinfandel has_color Red,
FormanCabernetSauvignon has_color Red,
GaryFarrellMerlot has_color Red,
StGenevieveTexasWhite has_color White,
WhitehallLaneCabernetFranc has_color White,
VentanaCheninBlanc has_color White,
SaucelitoCanyonZinfandel1998 has_color Rose,
SelaksSauvignonBlanc has_color White,
SchlossVolradTrochenbierenausleseRiesling has_color White,
SeanThackreySiriusPetiteSyrah has_color White, SelaksIceWine has_color White,
SchlossRothermelTrochenbierenausleseRiesling has_color Rose, StonleighSavignonBlanc has_color White,
StGenevieveTexasWhite has_color White,
SantaCruzMountainVineyardCabernetSauvignon has_color Rose,
.....

4. Step 4:Subject 찾기

STEP 3에서 추출된 트리플 집합_sub2에서 마지막에 들어오는 arg3을 만족하는 Subject를 찾는다. STEP 3에서 Predicate까지 체크를 하였으며, 트리플의 마지막 구성인 Object는 트리플 집합_sub2에 대하여 질의를 수

행한다. 해당 모듈은 다음과 같다.

```
select {SUBJECT}
 from {TRIPLE_SET_sub2}
where
{TRIPLE_SET_sub2.object=:ls_arg3}
```

최종적으로 트리플 집합_sub2에서 object 값과 ls_arg3를 비교하여 일치하는 트리플의 Subject가 트리플 처리 언어의 한 사이클을 통하여 얻어지는 결과 값이 된다.

```
select {SUBJECT}
 from Wine.triple_sub2
where Wine.triple_sub2.object=:White
```

STEP 3에서 만들어진 Wine.triple_sub2 트리플 집합에서 object 값이 White인 트리플의 Subject를 찾아서 Return하는 질의 내용이다.

```
CorbansPrivateBinSauvignonBlanc,CorbansSauvignonBlanc,SelaksSauvignonBlanc,StonleighSauvignonBlanc,FormanChardonnay,MountEdenVineyardEdnaValleyChardonnay,MountadamChardonnay,PeterMccoyChardonnay,FoxenCheninBlanc,VentanaCheninBlanc,MountadamRiesling,SelaksIceWine,CorbansDryWhiteRiesling,SchlossRothermelTrochenbierenausleseRiesling,SchlossVolradTrochenbierenausleseRiesling,CongressSpringsSemillon,KalinCellarsSemillon,CortonMontrachetWhiteBurgundy,PulignyMontrachetWhiteBurgundy,StGenevieveTexasWhite,
```

5. Step 5:조건검색에 관한 정의

트리플 한 사이클을 처리하기 위해서 STEP 1에서 STEP 4까지의 과정이 이루어지며, 이 과정은 조건검색 키워드 [AND, OR]에 의해서 반복적으로 이루어진다. 이러한 조건 검색의 수행으로 본 연구의 질의 언어를 사용하여 상세한 트리플 처리가 가능하다.

그림 1은 질의 처리 과정의 일련의 프로세스를 보여주고 있다. 컨트롤된 질의 처리 기반 온톨로지 탐색을 하기 위해서는 사람이 하는 일련의 사고처리 프로세스를 기계가 대신 수행할 수 있도록 설계해주어야 한다.

본 연구에서는 기계의 수행범위를 넓히고 사람의 수행 범위를 최소화하여 기계가 온톨로지를 탐색하고 추론하여 온톨로지 중심의 질의 처리가 가능하도록 하였다. 또한 온톨로지에 대한 효율적 질의 처리를 위하여 추론 규칙 기반 파싱을 통한 트리플 파일을 생성하는 부분과 이러한 트리플 파일에 대하여 효율적 질의가 가능한 질의 처리 부분을 통합하여 질의 처리 모듈을 설계하였으며, 기존의 질의수행 방식의 복잡함을 해결하고자 형식화된 질의 입력 방식을 채용하여 좀 더 쉽고 정확한 질의 처리가 가능하도록 하였다.

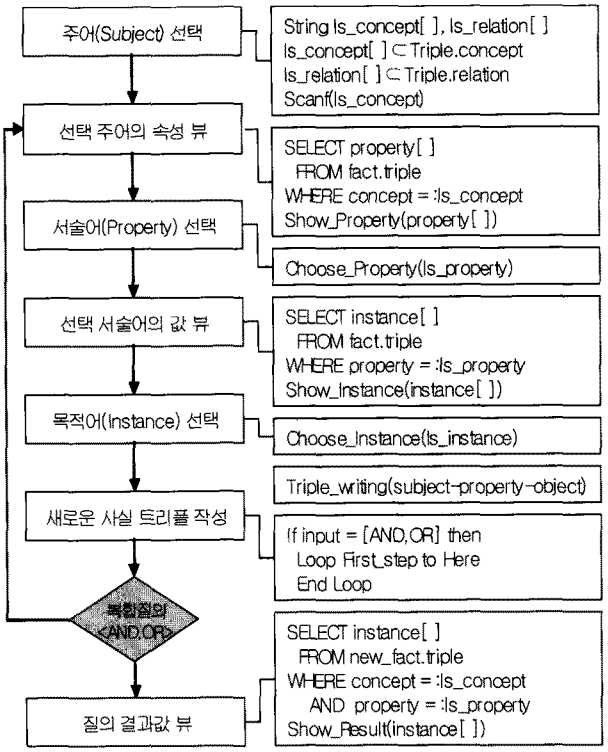


그림 1. 질의 처리 모듈의 수행 프로세스
Fig. 1. Performance Process of Query Process Module.

기존 대부분의 온톨로지 질의 처리 시스템에서는 단순한 온톨로지 질의 언어에 기반하여 온톨로지내의 구조적 사실들에 대한 질의 결과를 사용자에게 제공하고 있다. 그렇지만, 온톨로지의 사용 목적인 추론에 대한 결과를 제공하지 못하여 사용자에게 온톨로지 사용에 대해 어필하지 못하고 있다. 이에 반해, 본 연구의 질의 모듈은 추론 규칙 기반으로 얻어진 트리플 사실들을 사용하여 질의 처리가 이루어지고 있다.

IV. 실 험

본 연구의 질의 처리 방법론 평가를 위해 W3C에서 만든 와인 온톨로지(wine.rdf)를 실험 대상으로 내용 평

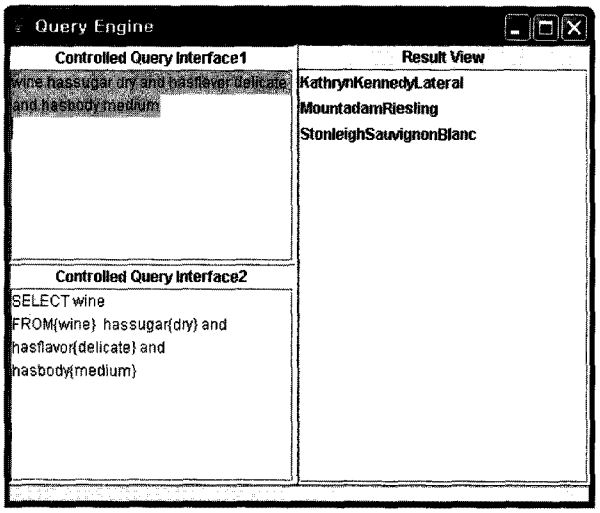


그림 2. 질의 처리 모듈 실행 화면
Fig. 2. Execution Picture of Query Process Module.

가를 수행하였다. 먼저 와인 온톨로지는 파싱을 거쳐 총 1505개의 트리플 사실들이 생성되었으며, 1505개의 트리플들은 온톨로지 언어의 구조적 특징에 기반한 Fact 트리플 798개와 온톨로지 어휘의 규칙에 기반한 Semantic 트리플 707개로 구성되어 있다. 생성된 트리플들에 기반하여 텍스트 기반의 질의 처리 인터페이스에서 질의를 수행하는 실행화면은 그림 2와 같다.

그림 2의 내용은 평가를 위해 테스트한 질의 내용에 대한 결과들을 보여주고 있다. 인터페이스 1과 같은 텍스트 형식의 자연어와 같은 질의 처리가 가능하며, 인터페이스 2에서는 기존의 질의 처리 방식의 구조적 형식을 보여주고 있다.

결과 화면에서 질의에 대한 결과들을 보여주며, 쉬운 결과 확인을 통하여 사용자들은 자신이 구축한 온톨로지에 대한 평가가 용이하게 된다. 결론적으로, 온톨로지 처리의 통합화와 쉬운 질의 처리 환경을 제공함으로써 온톨로지 사용의 범용성을 높이고, 질의 모듈의 효과로 온톨로지의 쉬운 평가를 통한 높은 활용성을 기대할 수 있다.

V. 결 론

기존의 온톨로지 질의처리를 위한 시스템 및 질의 언어들의 사용은 대부분 구조적 질의 형식(SELECT, FROM, WHERE)에 의존하여 사실상 질의를 수행하는 사용자는 질의 언어의 구조적 사용 방법 및 질의 수행을 위해 온톨로지 내용을 사전에 파악하고 있어야 온톨로지에 대한 질의 처리가 가능하였다. 이러한 질의 처리 환경에서 온톨로지를 연구하는 대부분의 연구자들은 쉽게 이러한 질의 시스템을 다룰 수 있겠지만 시맨틱 웹이 차세대 웹으로 자리매김하기 위해 이러한 연구자뿐만 아니라 일반사용자에게도 쉽게 온톨로지에 대한 질의를 수행할 수 있는 텍스트 기반의 질의처리 인터페이스를 구현하였으며 이를 활용하여 온톨로지의 범용성을 높였다.

제안된 질이 처리 시스템을 통해 사용자는 복잡한 질의 구문이나 온톨로지의 내용을 알지 못하더라도 쉽게 온톨로지 내용에 대한 질의 수행이 가능하도록 하였으며 사용자는 텍스트 창에 마치 자연어를 입력하는 것과 같은 질의 처리가 가능하도록 하였다.

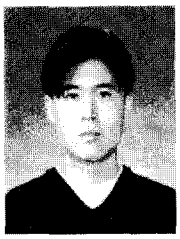
참 고 문 헌

[1] 윤여름 외 역, "시맨틱웹프라이머", 기파랑, 2007.
[2] Volker Haarslev, R.M.o., and Michael Wessel, Querying the Semantic Web with Racer + nRQL, In Proc. of the KI-2004 In tl. Workshop on Application of Description Logics (A이 04), 2004.
[3] Robin Cover, Hewlett-Packard Submits Query Language for RDF(RDQL) to W3C. 01-14, 2004.
[4] Andy Seaborne, from RDQL to SPARQL, 04-13, 2006.
[5] Mike Dean, G.G., Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider and Lynn Andrea Stein, OWL Web Ontology Language Reference. 2004.

저 자 소 개

김 원 필(정회원)

1994년 호남대학교 전산통계학과 학사 졸업.
1999년 호남대학교 컴퓨터공학과 석사 졸업.
2004년 조선대학교 전자계산학과 박사 졸업.
2008년~현재 조선이공대학 사이버보안과교수
<주관심분야 : 유비쿼터스, 정보검색, 정보보안>



공 현 장(정회원)

2002년 조선대학교 전자계산학과 학사 졸업.
2004년 조선대학교 전자계산학과 석사 졸업.
2007년 조선대학교 전자계산학과 박사졸업
2008년~현재 조선대학교 병원 전산실
<주관심분야 : 유비쿼터스, 정보검색, 정보보안>