

# 대용량 ABox에서 서술논리 SHIQ(D) 추론 지원 방법

(A Method for Supporting Description Logic SHIQ(D)  
Reasoning over Large ABoxes)

서 은 석<sup>†</sup>      최 용 준<sup>†</sup>      박 영 택<sup>††</sup>  
(Eun-Seok Seo)    (Yong-Joon Choi)    (Young-Tack Park)

**요 약** 현존하는 추론 엔진들은 대부분 Tableaux 알고리즘 기반의 TBox의 최적화를 위한 연구를 진행하였다. 하지만 현실에서 대용량의 ABox를 추론하기 위해서는 유한한 시간 내에 결정 가능하지 못하다. 따라서 실용성 있는 추론 엔진 효율을 위해서는 대용량 데이터를 가지는 ABox를 위한 최적화된 추론 기법이 필요하다.

본 논문에서는 OWL-DL 기반의 온톨로지(Ontology)를 데이터로그(Datalog)와 같은 규칙(Rule) 형태로 변형하여 관계형 데이터베이스와 같은 저장장치와 연동하기 위한 방법을 이용한다. 최종적으로 실세계의 환경에서의 데이터타입 속성(Datatype Property)이 포함된 SHIQ(D) 구성의 실용적인 지식 표현 시스템을 수행하고자 한다. 따라서 OWL이 가지는 공리(Axiom)를 이용한 데이터타입이 포함된 규칙을 적용한 추론 방법에 대해서 제안하였다.

**키워드** : 온톨로지, 규칙, 데이터 로그, OWL, Datatype, 추론

**Abstract** Most existing deductive engines study for optimization of TBox based on Tableaux algorithm. However, in order to deduce mass-storing ABox in reality, it can't be decided in finite time. Therefore, for the efficiency of the deductive engine, there needs to be reasoning technique optimized for ABox.

This paper uses the method that changes OWL-DL based Ontology to the form of Rule like Datalog in order to interlock store device such as RDBMS. Ultimately, it tries to in circumstance of real world. Therefor, using Axiom that OWL holds, it suggests reasoning method that applies rules including datatype.

**Key words** : Ontology, Rule, datalog, SHIQ(D), Datatype, Reasoning

## 1. 서 론

현존하는 대부분의 추론엔진들은 OWL-DL기반의 온톨로지(Ontology)를 이용한 추론을 수행한다. OWL-DL은 W3C표준의 표현력이 서로 다른 세 가지의 하위 언어들 중 하나이며 서술논리(Description Logic)[1]를 기반으로 한다. 서술논리의 구조는 개념(Concept)을 표현하기 위하여 온톨로지의 스키마 구조를 가지는 TBox와

개념에 포함되는 직접적인 실세계의 데이터 개체(Individual)들을 표현하고 있는 ABox로 구성된다. 이러한 구조를 바탕으로 하여 추론엔진은 명시적(Explicit)으로 표현된 지식으로부터 암묵적(Implicit)인 지식을 도출할 수 있다. 추론엔진들의 추론 처리 과정에서의 발생되는 시간 비용을 줄이기 위하여 다양한 최적화 기법이 제시되었으며 대부분의 추론엔진들은 TBox를 통한 최적화 기법에 초점을 두었다. 하지만 최근에 시맨틱웹(Semantic Web)[2]에 대한 연구가 진행되어 오면서 실용성있는 대용량의 ABox 처리를 위하여 기존 추론엔진들의 수행해 오던 Tableaux 알고리즘[3] 기반의 TBox 최적화를 통한 추론 수행으로는 유한한 시간 이내에 지식을 도출하는데 한계가 있다. 따라서 기존 추론 엔진과는 다른 추론 방안이 필요하며 궁극적으로 대용량 ABox를 위한 최적화 기법을 통한 실세계의 응용 어플리케이션에서

· 본 논문은 숭실대학교의 지원을 받았습니다.

† 학생회원 : 숭실대학교 컴퓨터학과  
erin214@ailab.ssu.ac.kr  
ai@ailab.ssu.ac.kr

†† 종신회원 : 숭실대학교 컴퓨터학과 교수  
park@computing.ssu.ac.kr

논문접수 : 2006년 7월 3일

심사완료 : 2007년 4월 9일

효율성 있는 연구 방향이 필요하다.

이러한 문제점들을 해결하기 위하여 Tableau 방식이 아닌 서술 논리 프로그램(Description Logic Program) 방식을 사용할 수 있다. 서술논리 프로그램 방식은 기존에 잘 알려진 사실(Fact)과 규칙(Rule)의 집합을 다루는 논리 프로그램(Logic Program)과 서술 논리를 결합한 형태를 말한다. 이러한 접근 방법은 OWL-DL 기반의 온톨로지를 선언적 데이터로그(Datalog)와 같은 규칙 언어와 결합하기 위한 방법이다. 데이터로그는 관계형 데이터베이스(RDBMS)에서 추론 규칙을 직접 표현할 수 있으며 규칙에 만족하는 결과에 대하여 빠른 시간 내에 찾을 수 있도록 한다. 최종적으로 온톨로지의 스카마 구조인 TBox를 규칙 형태로 변환하여 비교흡수(Resolution)방식을 수행하며 실질적 데이터인 ABox를 영구적인 저장소에 분리하여 대용량 데이터에 대한 실용적인 추론 수행이 이루어 질 수 있도록 하기 위한 방법이다. 하지만 이러한 규칙 형태로의 변환과정에서 수치적 연산을 위한 데이터타입이 포함된 SHIQ(D) 구성의 온톨로지는 비교흡수 기반의 정리 증명기(Theorem Prover)를 위하여 상당히 복잡하며 정교한 기술이 요구된다. 예를 들어 수치형 제약(Number Restriction)을 위한 처리를 위해서는 동치성(Equality)에 대한 추론이 요구되며 복잡한 계산이 필요로 하다. 따라서 데이터타입이 포함된 온톨로지를 처리하여 최종적으로 추론을 수행하기 위하여 새로운 접근 방식이 필요로 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로서 Tableau 알고리즘 기반의 접근방식과 선언적 데이터로그 접근방식이 가지고 있는 특징과 문제점에 대하여 서술하였다. 3장에서는 본 연구에서 사용되는 OWL-DL에서의 서술 논리에 대한 관계 및 SHIQ(D) 형태의 명칭 표기에 대해서 살펴보았다. 4장에서는 본 논문에서 제안하고 있는 수치적 연산을 위한 데이터타입이 포함되는 OWL-DL 기반의 온톨로지의 추론 수행 방법을 위한 데이터로그 형태로서 룰 형태로서 변환과정에 대하여 살펴보고 마지막으로 5장에서는 본 논문의 결론과 추후 진행 하고자 하는 연구 방향에 대하여 살펴보고자 한다.

## 2. 관련 연구

추론 엔진에 대한 설계 방법은 크게 두 가지 형태의 접근 방식을 가지고 있다. 첫 번째 방법은 Tableau 알고리즘 기반의 접근 방식이 있으며 FaCT/FaCT++[4], RACER[5], Pellet[6]과 같은 추론기(Reasoner)들이 있다. 이러한 접근 방법은 TBox 추론에서는 효율적으로 평가되었지만 대용량의 ABox 추론에서는 실용성이 떨어지는 것으로 평가되었다. 이러한 단점을 해결하기 위

하여 선언적 데이터로그와 같은 접근방식이 있으며 대표적으로 KAON2[7],[8]가 있다.

### 2.1 Tableau 알고리즘 기반의 접근 방식

Tableau 알고리즘 기반의 대표적 추론기인 RACER(Renamed ABox and Concept Expression Reasoner)는 SHIQ(D) 범위의 서술논리 문법의 추론을 수행 하도록 하기 위하여 개발된 추론 엔진이다. RACER는 장점으로는 Tableau 알고리즘 기반의 TBox 최적화[9]를 통하여 현존하는 대부분의 추론 엔진보다 TBox 추론에서는 뛰어난 성능을 나타내고 있다. 또한 각 개체가 가지는 수치적 형태의 추론을 위한 데이터타입 속성에 대하여 제한적으로 추론을 수행할 수 있다. 하지만 RACER가 가지는 단점으로는 대용량 ABox에서 Tableau 알고리즘의 한계를 나타내고 있다. 초기 시스템에 대한 설계 당시 ABox에 대한 실용적인 추론 수행을 목표로 두었지만 Tableau 알고리즘의 추론 수행방식으로 인하여 대용량의 ABox 추론에서는 KAON2와 같은 추론 엔진에 비해 효율성이 크게 떨어지는 것을 알 수 있다.

### 2.2 선언적 데이터로그(Disjunctive Datalog) 접근 방식

대표적 추론기인 KAON2는 OWL-DL로 표현된 대용량의 ABox에 대하여 기존의 추론엔진이 사용하고 있는 Tableau 알고리즘 방식이 아닌 선언적 데이터로그 언어를 통한 데이터베이스에서의 추론을 모두 처리할 수 있도록 설계된 추론 엔진이다. KAON2의 주요 장점으로는 작은 사이즈의 TBox와 대용량의 ABox일 경우 매우 효율적인 추론을 수행하는 것을 나타내었다. 하지만 KAON2 단점으로 OWL의 수치형 연산을 위한 데이터타입과 같은 구체화된 도메인에 대한 질의 수행 시 결과를 도출할 수 없는 단점이 있다. 예를 들어 'Person 클래스에서 age가 30이상인 모든 개체들을 검사하라'와 같은 질의응답을 수행 할 수 없다.

## 3. OWL-DL에서의 표현 범위

실세계 지식에 대하여 구조적 표현을 위해서 W3C의 표준화된 OWL의 세 가지 서브 언어 중 하나인 OWL-DL은 서술논리(Description Logic) 기반으로 모든 결론이 계산될 수 있는 특징인 계산학적 완전성(Computational Completeness)과 모든 계산은 유한한 시간 내에 끝낼 수 있는 결정가능성(Decidability)을 유지하면서 최대의 표현 범위를 제공해 준다.

지식의 표현과 공유를 위하여 1970년도부터 개발되어 온 서술논리는 실세계 환경의 도메인에 대한 지식에 대하여 표현하기 위한 언어로서 일반적인 개념에 대한 지식 표현 및 각 개념의 관계를 정의할 수 있다. 따라서 현실에 존재하는 방대한 양의 정보를 사람을 대신 기계

가 처리 가능하도록 하기 위한 추론을 통하여 새로운 의미 정보를 도출 할 수 있다. 서술논리를 이용한 지식 베이스는 그림 1과 같이 크게 두 개의 구성 요소로 이루어진다. 그 중 하나는 개념(Concept) 및 용어(Terminology)에 관한 지식을 포함하기 위하여 온톨로지의 스키마 구조를 가지는 TBox와, 다른 하나는 개념에 포함될 수 있는 직접적인 실세계의 데이터 개체들의 선언(assertion)에 관한 지식을 표현하고 있는 ABox로 구성된다. 이러한 구조를 바탕으로 구성된 지식베이스는 서술논리 언어를 이용하여 표현되며, 추론엔진은 명시적(Explicit)으로 표현된 지식으로부터 추론(Reasoning)을 통하여 암묵적(Implicit)인 새로운 지식을 도출할 수 있다.

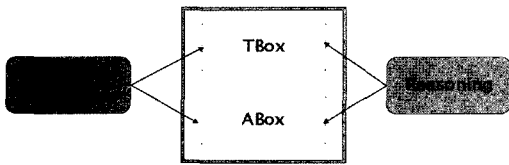


그림 1 Description Logic 구조

3.1 OWL-DL에서의 하위 구성 요소

서술논리 기반으로 구성된 OWL-DL 온톨로지 언어에서는 표현되는 범위에 따라서 SHOIN(D), SHOQ(D), SHIQ(D)와 같은 하위 구성 요소로 이루어 질 수 있다 [1]. 대부분의 추론 엔진들은 SHIQ(D) 범위의 표현을 통하여 추론을 수행하며 SHOIN(D)으로 갈수록 표현하는 문법이 증가하게 됨으로 구현 하게 되는 알고리즘의 복잡도로 시간 비용 또한 증가하게 된다.

서술논리 언어에서 사용되는 표현의 구성에 따라 해당되는 구성 기호를 통하여 명칭을 부여한다. 그림 2에서는 SHOIN(D), SHOQ(D), SHIQ(D)와 같은 하위 구성 요소에 사용되는 명칭에 대한 정리를 나타낸다.

서술논리의 가장 하위 요소인 AL(Attribute Language)은 부정의 의미를 나타내는 negation을 추가하여 ALC로서 나타내며 구성요소로는 합집합(disjunction), 교집

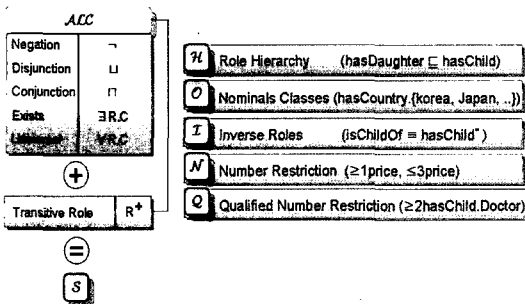


그림 2 OWL-DL이 가지는 표현 범위

합(conjunction), 존재정량자(Exists quantifier), 전체정량자(Universal quantifier)로 구성된다. ALC에 이행적 속성으로 사용되는 Transitive Role을 추가하여 S라는 명칭을 부여한다. H는 속성의 계층관계에 대한 표현 여부를 의미하며 hasChild의 하위 속성으로 hasDaughter와 같은 표현이 가능한 Role Hierarchy를 나타낸다. O는 Nominals Classes로 속성이 가지는 레인지를 클래스에 포함되는 고유 개체로 정의 할 수 있으며 OWL에서의 owl:oneOf와 같은 공리를 의미한다. I는 임의의 개념이 가지는 속성이 역으로 사용될 수 있음을 의미하며 그림에서의 예와 같이 isChildOf라는 속성은 hasChild의 역속성과 동치로 표현할 수 있음을 나타내는 Inverse Roles이다. N은 Number Restriction으로 임의의 개념이 가지는 수치적인 속성 값을 나타내며 Q는 속성 값의 범위를 정해주는 Qualified Number Restriction이다. 이러한 각 표현 능력을 조합하여 추론 엔진의 표현 범위에 대하여 명칭을 통한 간단한 구성이 가능하도록 할 수 있다. 본 논문에서는 SHIQ(D) 범위의 표현에서의 추론 수행을 목적으로 하고 있다.

일반적으로 SHIQ 구성의 추론 수행에 있어서 ExpTime에서 계산이 완료될 수 있는데 Tableaux 알고리즘 방식은 SHIQ 구성의 추론 수행에서 2NExpTime이 소요되기 때문에 높은 시간 복잡도를 가지게 된다. 따라서 추론엔진들의 추론 처리 과정에서의 발생하는 시간 비용을 줄이기 위하여 추론 알고리즘의 다양한 최적화 기법[4,9]이 제시 되었으며 대부분의 추론엔진들은 TBox를 통한 엔진의 최적화 기법에 초점을 두었다. 하지만 방대한 자원에 대한 추론 수행을 위해서 TBox의 최적화만으로는 대용량의 ABox의 추론 처리에 한계가 있다. 예를 들어 지식베이스에 표현된 개념에 대한 포한 관계인 CUD가 만족하는지에 대한 여부를 결정하기 위하여 TBox에 해당되는 임의의 ABox에 대하여 ABox에서의 모순(Contradiction)이 발견될 때까지 일관성(Consistency) 검사를 수행하여야 한다. ABox의 사이즈가 작을 경우에는 유한한 시간 이내에 추론을 통한 새로운 지식을 도출 할 수 있다. 하지만 실세계의 온톨로지 모델링에 대하여 발생하는 대용량 ABox를 추론[8]하기 위해서는 다른 접근 방식의 최적화 기법에 대하여 생각해 보아야 할 필요성이 있다.

4. 대용량 ABox에서의 데이터타입 지원 방안

본 논문에서는 OWL-DL 기반의 온톨로지를 선언적 데이터로그와 같은 규칙 언어로의 변환 시 SHIQ(D) 범위의 데이터타입에 대한 지원 방법을 제안하였다. 데이터타입이란 개념은 구체화된 도메인(Concrete domain)을 의미한다. OWL-DL에서 데이터타입은 XML 스키마

[10]로 표현될 수 있으며 string, integer 등으로 각 개체가 가지는 속성 값을 정의 할 수 있다. 하지만 이러한 규칙 형태로의 변환과정에서 수치적 연산을 위한 데이터타입이 포함된 SHIQ(D) 구성의 온톨로지는 비교흡수기반의 정리 증명기를 위하여 상당히 복잡하며 정교한 기술이 요구된다. 예를 들어 수치형 제약을 위한 처리를 위해서는 동치성에 대한 추론이 요구되며 복잡한 계산이 필요로 하다.

4.1 시스템 구조

본 논문에서 제안하는 방식에 대한 그림 3에서 시스템 구조를 묘사하였다. Protege와 같은 온톨로지 모델링 툴을 이용하여 온톨로지를 모델링 하면 모델링된 온톨로지 파일을 Concrete Domain Preprocessing 모듈에 전달한다. Concrete Domain Preprocessing 모듈은 Datatype이 포함되는 지식베이스에 대한 분리작업을 수행한다. 이러한 분리작업을 수행하는 이유는 최종실험을 위하여 SHIQ구성 Knowledge Base는 기존 Kaon2 알고리즘을 이용하며 Datatype이 포함되는 KB는 본 논문에서 제시하는 Datatype Handling 알고리즘을 이용하기 위함이다. Ontology Clausification 모듈에서 OWL-DL의 스키마 구조를 가지는 TBox에 대하여 FOL 문법으로 변환하며 최종적으로 Datalog Engine에서 사용되는 룰 처리를 위한 Horn clauses 형태로 변환된다. 본 시스템에서의 Datatype에 대한 처리 지원 방안을 위하여 Datatype Rule Handler는 Datatype이 포함되는 TBox에 대해서 Datalog Engine에 사용되는 Horn Clauses 형태로 변환된다. 최종적으로 Ontology Clausification 모듈과 Datatype Rule Handler 모듈에서 추출되는 KB는 Datatype이 포함되는 SHIQ(D)의 구성을 가지게 된다.

그림 4에서는 데이터 타입 지원을 위한 트리플 구조에 대하여 설명하고 있다. 그림의 예에서와 같이 'HighEnd 클래스가 가지는 속성인 hasPrice는 2000 이상이다.' 라는 정의에 대하여 서술논리로 HighEnd ≡  $\exists$  hasPrice.  $\geq$  2000 과 같이 표현 될 수 있다. OWL-DL로서 표현할 경우 rdf:ID를 이용하여 HighEnd 클래스를 정의한다. HighEnd 클래스가 가지는 hasPrice라는 속성 정보에 대하여 owl:Restriction 공리를 통하여 hasPrice는 owl:DatatypeProperty 이며 owl:minCardinality 공리로서 최소 2000 이상으로 정의될 수 있다.

모델링된 온톨로지를 데이터타입 규칙처리기로 전달하기 위하여 전처리 통하여 트리플 구조로 변환한다. 트리플 구조로 변환된 온톨로지는 그림 6과 같이 클래스와 속성에 대한 타입과 임의의 클래스가 가지는 서브클래스 정보들을 트리플 구조로서 알 수 있다. 예를 들어 <HighEnd><type><Class>는 'HighEnd 클래스의 타입은 클래스이다.'라는 사실을 의미하며 <HighPrice><type><DatatypeProperty>는 'hasPrice의 타입은 데이터타입 속성이다.' 라는 사실을 의미한다. 임의의 클래스에 제약 정보를 부여할 경우 그림에서와 같이 restrictionID가 트리플 구조에서 생성될 수 있다. 예를 들어 'HighEnd 클래스는 hasPrice로서 최소 2000이상의 값을 갖는다' 라는 제약 정보를 부여할 경우 HighEnd 클래스는 익명의 클래스(restrictionID)에 하위클래스이고 이러한 <restrictionID>의 값을 갖는 익명 클래스는 <onProperty><minCardinality><type>등의 OWL이 가지는 공리 정보를 통하여 임의의 클래스에 대한 제약 정보를 알 수 있다.

본 논문에서 이용되는 데이터로그는 연역적 데이터베이스(Deductive Databases)인 fdnl한 질의와 규칙 연

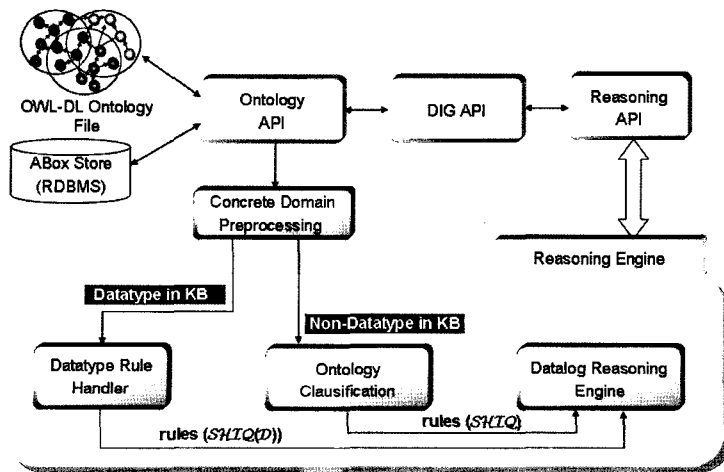


그림 3 시스템 구조

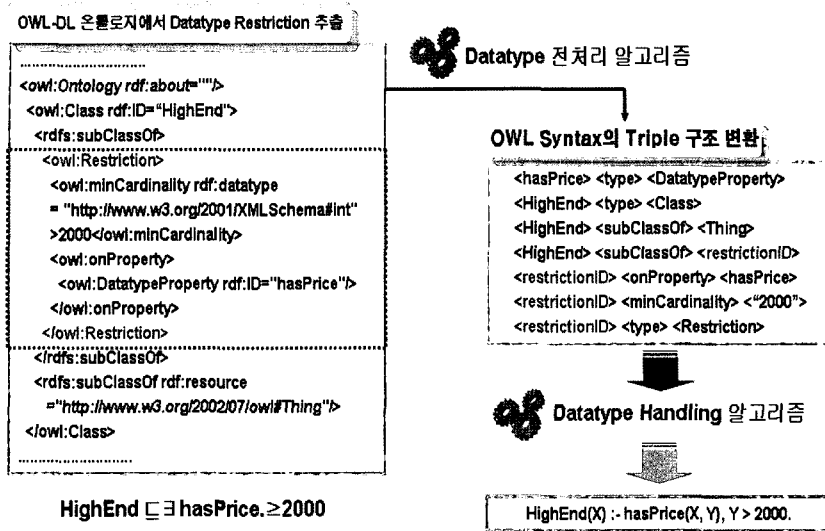


그림 4 OWL 트리플 구조 적용

어이며 문법적으로는 프로그래밍과 유사한 형태이다. 데이터타입을 사용한 질의 평가에서 정당(Sound)하고 완전(Complete)하며 대용량 데이터베이스에서 효율적으로 사용할 수 있다.

4.2 데이터타입 전처리

Datatype 전처리 모듈에서는 온톨로지 스키마로부터 Datatype 속성을 공유하는 개념과 개념간의 관계를 따로 분리하여 Datatype 규칙 처리기에 전달하도록 한다. 이러한 작업 수행을 위해서 Datatype 전처리 모듈에서는 전달받은 OWL 온톨로지의 TBox에서 datatype을

추출하기 위하여 온톨로지가 주어(Subject), 서술어(Predicate), 목적어(Object) 형태의 트리플 형식으로 변환하도록 하였다. 예를 들어 'LargeMonitor는 Monitor이면서 hasScreenSize 속성으로 20이상의 값을 갖는다.'라는 명제에 대하여 그림 5와 같이 온톨로지 문법으로 정의 될 수 있다.

정의된 문법에 대하여 OWL 트리플 구조로 나타내어지게 되면 그림 6에서와 같이 나타낼 수 있다. 20이하의 속성을 정의하기 위하여 OWL의 minCardinality 공리를 정의할 수 있으며 LargeMonitor라는 클래스가 가지

```

<owl:Class rdf:ID="largeMonitor">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</owl:minCardinality>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:about="#hasSize"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Monitor"/>
</owl:Class>
  
```

그림 5 Datatype이 포함된 온톨로지 문법

```

<LargeMonitor> <rdfs:subClassOf> <Monitor>;
<LargeMonitor> <rdfs:subClassOf> <Restriction ID>;
<LargeMonitor> <rdf:type> <owl:Class>;
<Restriction ID> <rdf:type> <owl:Restriction>;
<Restriction ID> <owl:onProperty> <hasScreenSize>;
<Restriction ID> <owl:minCardinality> <"20"^^http://www.w3.org/2001/XMLSchema#int>;
  
```

그림 6 트리플 형태로 변환된 온톨로지

는 추가적 제약을 표현하기 위하여 그림에서와 같이 Restriction ID를 기준으로 하여 Datatype의 속성 정의와 속성 값이 가지는 실제 값에 대하여 XML Schema로서 표현 할 수 있다.

4.3 데이터타입 규칙 처리기

Datatype 규칙 처리기에서는 Datatype 전처리 모듈에서 트리플 형식으로 추출된 KB를 Datalog Engine에서 사용하도록 하기 위한 Rule 형태로의 변환과정을 수행한다. 이러한 작업 수행을 위하여 변환된 트리플 구조에서 Datatype이 가지는 Restriction ID를 추출 할 수 있다. 본 논문에서는 이러한 Restriction ID가 가지는 관계정의를 통하여 Datalog 형태의 룰을 구성하였다. 그림 7에서는 KB에서 Datatype을 위한 룰의 구성을 위한 알고리즘의 예를 나타내고 있다. Datatype 전처리 모듈로부터 전달 받은 트리플 형태의 KB에서 최소값이나 최대값에 대한 Datatype 처리를 위하여 트리플 KB 상에 이러한 정보를 나타내고 있는 owl:minCardinality 또는 owl:maxCardinality에 대한 정의가 있는지에 대한 검사를 수행한다. 예를 들어 minCardinality가 KB상에 존재한다면 Restriction ID 및 클래스가 가지는 추가적 속성에 대한 체크를 수행한다. minCardinality 처리를 위한 모든 조건이 만족되면 Restriction ID를 기준으로 서브클래스 정보는 Datalog 룰의 Head 부분으로 나머지 제약 정보에 대해서는 룰의 Body 부분으로 구성되어 지게 된다.

최종적으로 변환되어진 Datalog 룰은 Datatype이 포함되지 않은 KB와 결합함으로써 완전한 SHIQ(D) 구성의 Datalog 룰로서 구성된다.

4.4 실험

데이터타입 문법이 포함되어져 있는 컴퓨터 도메인의 동일한 온톨로지에 대하여 본 시스템과 KAON2에서 처리해 보았다. 본 논문에서 제시하는 수치형 데이터타입의 비교 실험에 앞서 일반적인 질의에 대한 수행이 정상적으로 이루어지는가에 대하여 확인하며 이후에 수치형 데이터타입에 대한 비교 실험을 진행한다.

실험을 위하여 시스템의 각 단계별 과정을 나타낼 수 있는 시뮬레이터를 구현 하였다. 그림 8에서와 같이 본 시스템에서 제안한 처리 결과로 데이터타입 속성이 포함된 규칙과 사실에 대하여 나타낼 수 있었지만 KAON2에서는 수행할 수 없었다.

그림 9에서와 같이 비교가 아닌 명시적으로 주어진 수치값에 대한 결과를 반환 받을 수 있는지를 확인하기 위한 질의인 "monitor(X), hasScreenSize(X, 14)"에 대하여 정상적으로 수행이 되는 모습을 볼 수 있다.

더불어 명시적 클래스에 대한 멤버 객체를 반환받기 위한 OWL 공리(Axiom)를 이용한 질의에 대한 결과도 그림 10과 같이 수행되는 모습을 확인할 수 있다. 질의는 "monitor(X)"로 주어 monitor 클래스에 속한 모든 멤버 객체를 요구한다. 이에 대한 결과로 monitor 클래스에 속한 4개의 모니터 객체를 반환한다.

다음으로 데이터타입 속성이 포함되어진 온톨로지를 데이터로그의 규칙 형태로 변환하여 올바른 질의 여부가 수행되는지를 평가하기 위하여 다음과 같은 질의를 수행하여 보았다. 아래의 데이터타입 규칙과 사실이 있는 경우 smallMonitor 클래스가 가지는 모든 개체를 검사하고자 하였다.

데이터타입 규칙 1:

smallMonitor(Monitor):- monitor(Monitor),

```
[Input] C: Ontology KB Model
      [D]: Ontology Triple Mapping Set

[Output] Datalog Rule 형태

[D] = read(C)
For #=1...|D| do
  If (p == owl:minCardinality) then
    minCardinality_Process()
  Else if
    maxCardinality_Process()
End for

Function minCardinality_Process()
  If (owl:Restriction and owl:onProperty) then
    RDFNode <- owl:Restriction
    RDFNode <- owl:onProperty
    RDFNode <- owl:restrictionID
  End If

  HEAD <- HeadOfRule(owl:subClassOf, restrictionID)
  Body[] <- PropertyOfRule(owl:onProperty, restrictionID)
           , RangeOfProperty(owl:minCardinality, RangeClass)
End function
```

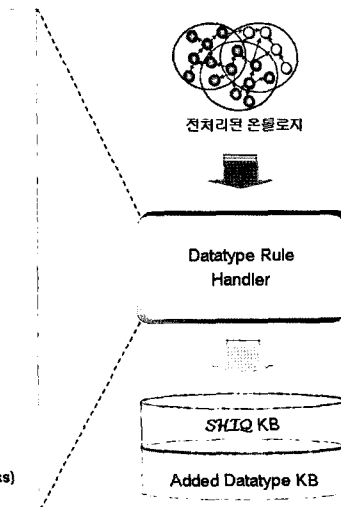


그림 7 Datalog 룰 구성을 위한 알고리즘

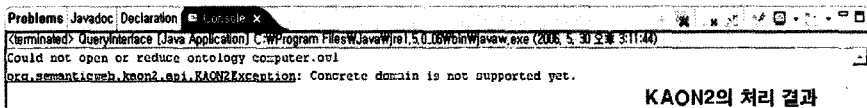
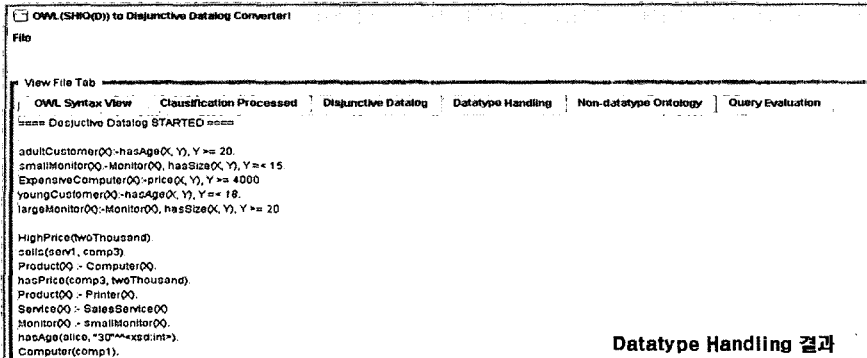


그림 8 SHIQ(D) 구성의 데이터로그 규칙 형태로 변환

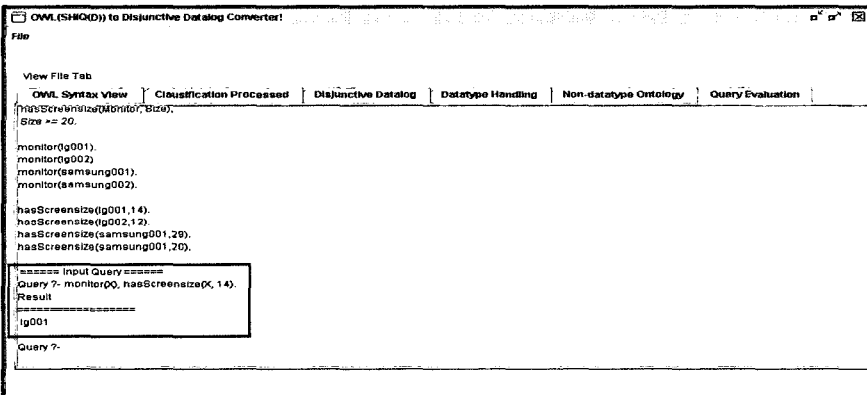


그림 9 명시적 수치 데이터에 대한 질의 결과

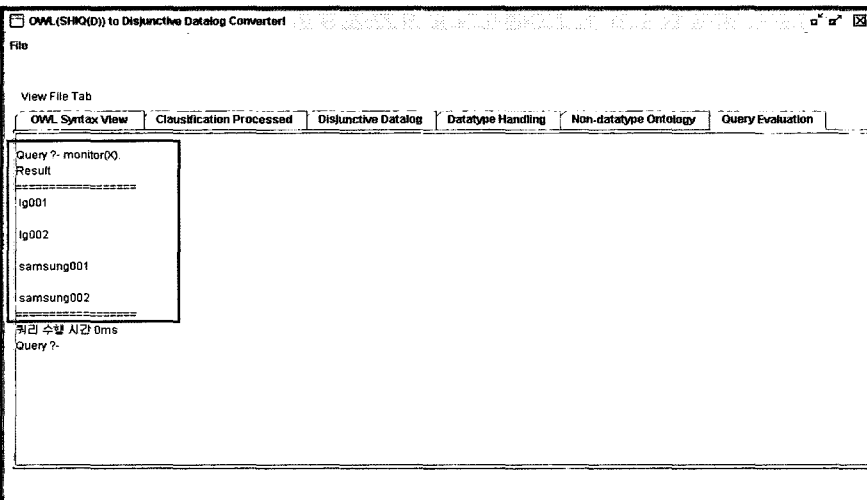
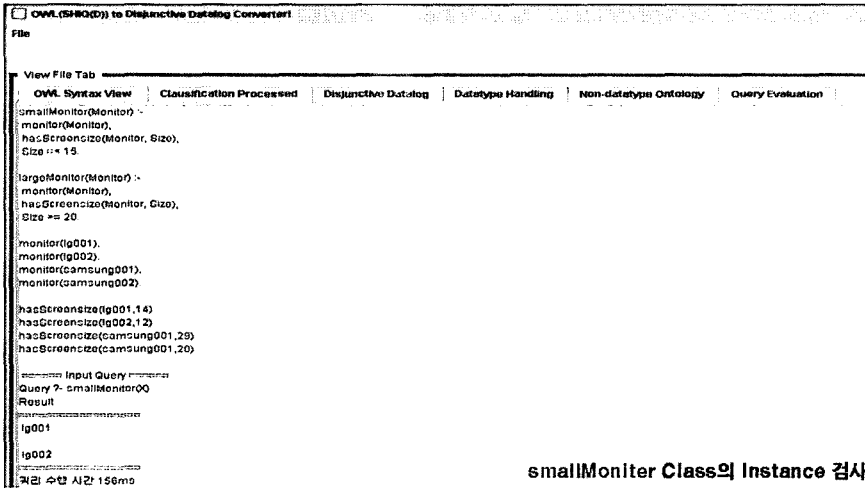


그림 10 멤버 객체를 반환하기 위한 질의 결과



smallMonitor Class의 Instance 검사

그림 11 질의 평가

hasScreenSize(Monitor, Size),  
Size =<15.

**데이터타입 규칙 2:**

largeMonitor(Monitor):- monitor(Monitor),  
hasScreenSize(Monitor, Size),  
Size =>20.

**사실 :** monitor(lg001), monitor(lg002), monitor(samsung001),  
monitor(samsung002),  
hasScreenSize(lg001, 14), hasScreenSize(lg002, 12)  
hasScreenSize(samsung001, 29),  
hasScreenSize(samsung002, 20)

**질의 :** smallMonitor(X).

그림 11의 질의 예에서 smallMonitor 클래스가 가지는 개체에 대한 검사 시에 규칙1을 통하여 monitor 클래스의 개체들 중 lg001과 lg002가 검색되는 것을 확인할 수 있었다.

**5. 결론 및 향후 연구**

본 논문에서는 Concrete Domain이 포함되는 SHIQ(D) 구성의 대용량 ABox에 대한 처리 방법에 대하여 제안하였다. 이러한 방법을 통하여 Datatype이 포함되는 OWL-DL 문법의 온톨로지를 Disjunctive Datalog의 Rule 형태로 변환하여 RDBMS와 같은 저장소에 ABox 지식을 저장하여 실용성 있는 추론이 가능하도록 하였다. 향후 연구로서는 좀 더 큰 범위에 대한 Datatype 처리 방법 및 SparQL[11]과 같은 표준화된 질의응답을 위한 문법 적용에 대하여 연구를 진행 하고자 한다.

**참 고 문 헌**

[1] F. Baader et al., 2003, "The Description Logic Handbook: Theory, Implementation and Applications," Cambridge University Press.

[2] G. Antoniou and F. van Harmelen, 2004, "A Semantic Web Primer"

[3] Ian Horrocks and Ulrike Sattler. A tableaux decision procedure for SHOIQ. In Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005), 2005.

[4] I. Horrocks and U. Sattler. Optimised reasoning for SHIQ. In Proc. of the 15th Eur. Conf. on Artificial Intelligence (ECAI 2002), pp. 277-281, 2002.

[5] V. Haarslev and R. Moller. RACER User's Guide and Reference Manual Version 1.9. <http://www.racer-system.com/products/racerpro/users-guide-1-9.pdf>

[6] <http://www.mindswap.org/2003/pellet/>

[7] B. Motik and U. Sattler, "Practical DL Reasoning over Large ABoxes with KAON2. Submitted for publication," 2005.

[8] U. Hustadt and B. Motik and U. Sattler, "Reasoning for Description Logics around SHIQ in a Resolution Framework," 2005.

[9] Dmitry Tsarkov and Ian Horrocks. Optimised classification for taxonomic knowledge bases. In Proc. of the 2005 Description Logic Workshop

[10] <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

[11] <http://www.w3.org/TR/rdf-sparql-query/>

[12] Horrocks and U. Sattler, "Ontology reasoning in the SHOQ(D) description logic," In B. Nebel, editor, Proc. of IJCAI-01, pages 199 -204. Morgan Kaufmann, 2001.

[12] <http://www.w3.org/TR/owl-semantics/>



[13] <http://www.w3.org/TR/owl-xmlsyntax/>



#### 서 은 석

2004년 2월 한국교육개발원 전자계산학(학사). 2006년 8월 숭실대학교대학원 컴퓨터학과 졸업(석사). 2006년 8월~현재 숭실대학교대학원 컴퓨터학과 박사과정 관심분야는 유비쿼터스 컴퓨팅, 시맨틱 웹, 추론, 온톨로지 등



#### 최 용 준

2004년 2월 한성대학교 공과대학 컴퓨터공학부 졸업(학사). 2006년 8월 숭실대학교대학원 컴퓨터학과 졸업(석사). 2006년 7월~현재 주식회사 니츠(기술개발부 전임연구원). 관심분야는 유비쿼터스 컴퓨팅, 시맨틱 웹, 온톨로지 추론, 다중 에이

전트 시스템 등



#### 박 영 택

1978년 서울대학교 전자공학과 졸업  
1980년 KAIST 전산학 석사 학위 취득  
1992년 University of Illinois at Urbana-Champaign 박사 학위 취득. 1981년~현재 숭실대학교 컴퓨터학부 교수. 관심분야는 시맨틱웹, 온톨로지 추론, 유비쿼터스 컴퓨팅, 개인화

에이전트, 기계학습, 전문가 시스템, Conceptual Clustering 등