OntoFrame®의 추론 시스템 개선

김평O, 이승우, 성원경
한국과학기술정보연구원 정보기술개발단
{pyungO, swlee, wksung}@kisti.re.kr

The Improvements in Reasoning System of OntoFrame®

Pyung KimO, Seungwoo Lee, Won-Kyung Sung
Korea Institute Of Science and Technology Information,
Information Technology Development Division

요 약

연구자의 R&D 전주기 과정을 지원하기 위해 KISTI에서 2006년 개발된 OntoFrame®은 과학기술 온톨로지와 DBMS 기반 추론 시스템을 적용하여 추론 서비스를 제공하였다. 2007년 개발중인 OntoFrame®의 추론 시스템은 대량의 인스턴스를 대상으로 빠른 추론 성능을 제공하기 위해서 저장 구조 변경 및 추론 프로세스 변경을 통해 추론 시스템의 성능을 개선하는데 중점을 두고 있다. 본 연구에서는 추론 시스템의 개선을 위해서 RDF 트리플의 데이터 구조 및 테이블 구조 변경, EMPV(Expanded Multiple Properties View) 개발 및 도입, 추론 서비스와 추론 시스템의 프로세스 변경, 저장 구조 관리기의 개발 연구를 수행하였다. 데이터 구조 및 테이블 구조 변경을 통해서 저장 공간의 오버헤드를 개선하였고, EMPV의 도입을 통해서 트리플 저장 구조의 비효율적인 자료 접근 방식을 개선하였으며, 프로세스 개선을 통해 추론에 소용되는 시간을 단축하였다. 또한 관리기 도입을 통해서 사용자 편의성 및 추론 시스템의 관리 기능을 제공하였다.

1. 서론

연구자의 R&D 전주기 과정을 지원하기 위해서 KISTI에서는 2006년 OntoFrame®을 개발하였다. OntoFrame®은 IT 분야 시소러스와 과학기술 온톨로지, 추론 시스템을 사용하여 연구자 네트워크, 연구 맵, 연구자 정보 등의 추론 서비스를 제공한다[1,2,3]. 연구성과물(논문, 특허, 보고서)과 기반 자원(인력, 기관, 주제, 분야 등)은 RDF 트리플(subject-predicate-object) 형태로 표현되고, 추론시스템은 사용자 정의 추론 규칙과 온톨로지 스키마 정보를 사용해서 트리플을 확장한 후 DBMS에 저장한다. 추론 서비스는 SPARQL[4]을 사용해서 추론 시스템에 질의를 전달하고, 추론 시스템은 DBMS를 사용해서 추론 서비스에 적합한 결과를 생성한 후 XML 형태로 변환하여 추론 서비스에 반환한다. OntoFrame®의 추론 시스템은 추론 서비스를 지원하기 위한 자료의 전처리 과정은 물론, 트리플에 대한 추론, 추론 결과를 추론 서비스에 적합한 형태로 가공하는 후처리 모듈을 모두 포함하고 있다[5].

2007년에는 OntoFrame®에서 제공하는 추론 서비스가 다양해지고 처리해야 할 인스턴스의 수가 급격히 증가함에 따라, 추론 시스템의 저장 구조 및 프로세스 개선을 통해 추론 성능을 높이는 것이 요구되었다. 본 연구에서는 추론 시스템이 대량의 트리플을 대상으로 빠른 추론이 가능하도록 저장 구조 및 프로세스를 개선하는 것에 중점을 두고, 다음과 같은 연구를 수행하였다.

- 데이터 및 테이블 구조 개선
- EMPV(Expanded Multiple Properties View) 개발
- 추론 프로세스 개선
- 추론 시스템 관리기의 개발

2장에서는 대량의 정보 처리를 위해서 저장 구조로 DBMS를 사용하는 추론 엔진의 성능 평가는 물론, 트리플 기반 저장 구조의 비효율적인 자료 접근 방식의 개선을 위한 연구에 대해서 기술하겠다. 3장에서는 OntoFrame®의 시스템 구성, 추론 시스템의 구성 요소 및 요소 별 기능, 추론 시스템의 저장 구조 개선 내용에 대해서 기술하고, 4장에서는 추론 프로세스 개선에 대하여 기술하겠다. 5장에서는 결론 및 향후 연구에 대해서 기술하겠다.

2. 관련 연구

추론 시스템이 실제 서비스에 사용되기 위해서는 대용량의 인스턴스 관리는 물론, 빠른 추론 속도가 보장되어야 한다. 실제 서비스에 사용되는 온톨로지는 온톨로지 자체가 복잡한 경우보다는 간단한 온톨로지에 많은 인스턴스를 가지는 것이 일반적인 경우에 속한다. LUBM에서도 실제 서비스에 적합한 추론

엔진을 판별하기 위한 벤치마킹 테스트로써, 복잡한 온톨로지와 적은 수의 인스턴스가 아닌 비교적 간단한 온톨로지와 대량의 인스턴스를 대상으로 추론 엔진간의 성능 비교 실험을 수행하였다[6]. LUBM의 실험 결과를 보듯이 대량의 인스턴스를 처리하기 위해서 대부분의 추론 엔진들이 DBMS를 저장 구조로 사용하고 있지만[7,8,9] 그 성능은 실제 서비스에 활용되기에 적합하지 못한 수준이다.

일반적으로 추론 엔진에서 사용되는 트리플 구조는 추론에는 적당하지만, 하나의 subject에 대한 속성이 분산되어 관리되기 때문에 하나의 subject와 관련된 2개 이상의 predicate 값을 참조하는 경우에는 비효율적이다. 즉 하나의 subject가 다수의 predicate를 가지는 경우 각각의 predicate 별로 별도의 테이블에 저장되기 때문에, 2개 이상의 predicate 값을 참조하기 위해서는 테이블 조인의 회수가 늘어나고, 이에 따라 성능이 저하될 수 밖에 없다. 이 문제를 해결하기 위해서 Jena2에서는 RDF 저장과 검색 연구를 통해 특정 속성과 관련된 subject-object 쌍을 별도의 속성 테이블(property table)로 생성하고 통합 관리함으로써 자료 접근의 효율성을 높였다[10]. Jena2에서는 속성 테이블로 관리되는 predicate에 대해서는 별도의 트리플로 관리하지 않음으로써 저장 공간을 효율적으로 사용하고자 하였다. 오라클은 효율적인 SQL 기반 RDF 질의 구조 연구를 통해 subject-property matrix materialized join views(SPMJVs)라는 개념을 도입하여 데이터에 대한 빈번한 조인 회수와 자료 접근에 따른 비효율성을 회피하였다[11].

3. 추론 시스템

OntoFrame®은 연구자의 전주기적 R&D 활동을 효과적으로 지원하기 위해서 URI 기반의 연구 성과물 및 기반 자원 관리, DBMS 기반의 추론 기능을 제공하고 있다. 그림 1과 같이 OntoFrame®은 4개의 요소로 구성되어 있다.

■ 자원

연구 성과물(논문, 보고서, 특허)과 기반 자원(인력, 기관, 주제, 분야), 과학기술 용어 시소러스 및 온톨로지(과학기술 온톨로지, 응용 온톨로지, 개인 온톨로지)를 포함한다. 연구 성과물과 기반 자원은 과학기술 온톨로지를 기반으로 하며, 사용자가 생성하는 Annotation과 같은 인스턴스들은 응용 온톨로지를 기반으로 생성된다. 개인 인톨로지는 개인 별 맞춤 정보 페이지를 생성하는데 사용된다. 사용되는 모든 자원은 식별 과정을 통해 고유한 URI를 부여 받는다.

■ UI

추론 서비스를 제공하기 위한 사용자 UI와 추론 시스템의 관리를 위한 관지라 UI로 구성된다. 사용자 UI를 통해 사용자는 Annotation 작업과 맞춤 페이지 생성을 위한 개인 프로파일을 작성할 수 있다. 관리자 UI를 통해 관리자는 추론 서비스를 지원하기 위한 사용자 정의 규칙을 편집과 추론 시스템 관리를 위한 관리 작업을 수행할 수 있다.

■ URI 서버

모든 자원에 URI를 부여하고 관리하며, 추론에 활용하기 위한 트리플을 생성하는 역할을 수행한다. 객체별로 각기 다른 URI 관리 체계를 가지며, 동명이인의 문제 해소를 위해 반드시 필요하다.

■ 추론시스템

추론에 필요한 정보의 트리플 저장, 관리는 물론 추론을 통해 추론 서비스에 적합한 추론 결과를 생성한다.

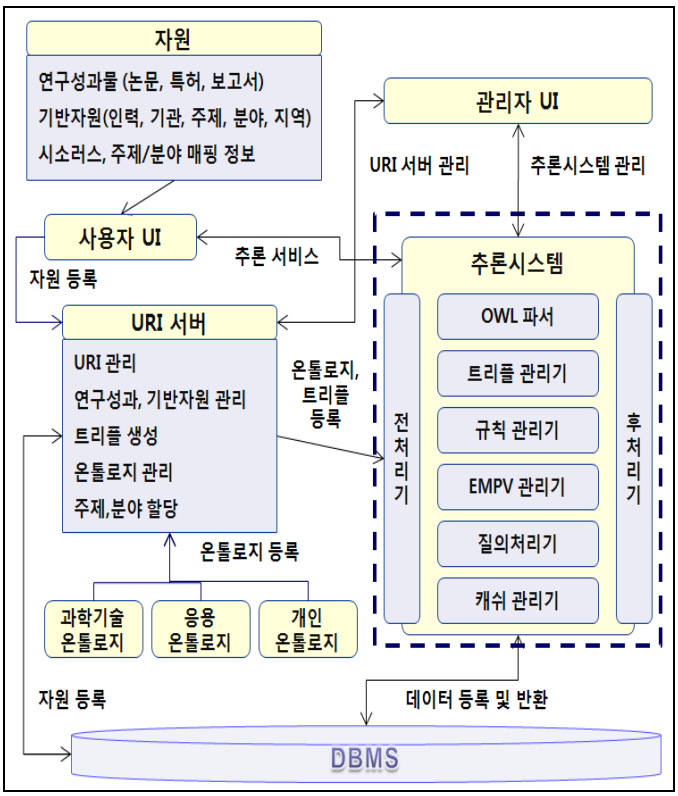

그림 1. OntoFrame® 구성

3.1 추론 시스템 구성

추론 시스템을 구성하는 구성 요소들은 다음과 같은 기능들을 수행한다.

■ OWL 파서

OWL 문서를 로딩하고, OWL 문서에 기술된 온톨로지 스키마를 트리플 형태로 변환하는 기능을 제공하며, 온톨로지의 유효성 검증도 수행한다.

■ 전처리기

인스턴스 트리플을 대상으로 추론 서비스에 필요한 정보를 트리플에 반영하기 위한 전처리 과정을 수행한다. 예를 들면, 연구 성과물의 중요도와 연구자의 저작 순위에 따라 저자 별 가중치를 계산하여 트리플에 반영하는 작업을 수행한다.

■ 트리플 관리기
온톨로지 스키마 트리플과 인스턴스 트리플의 유효성 검증, 트리플의 object 자료형에 적합한 테이블 생성, 트리플 저장 기능을 제공하며 질의 처리기에서 요청된 조건에 적합한 트리플을 검색하는 기능도 수행한다.
■ 규칙 관리기
관리자의 규칙 생성 및 관리를 지원하며, 생성된 규칙을 적용해서 지식을 확장하는 기능을 제공한다.
■ EMPV(Expanded Multiple Properties View) 관리기
특정 객체를 중심으로 다중 속성 관계를 자동을 설정하고 이를 사용해서 추론 성능을 높이는 기능을 제공한다.
■ 질의처리기
추론 서비스에서 요구하는 정보가 표현된 SPARQL 질의를 SQL 질의로 변환하고 캐쉬 관리기나 트리플 관리기를 통해 질의에 적합한 정보를 생성하여 반환하는 기능을 수행한다.
■ 캐쉬 관리기
추론 서비스의 응답 속도 개선을 위해 최근 사용된 질의와 결과를 관리하는 기능을 제공한다.
■ 후처리기
추론 서비스에 적합한 결과를 생성하기 위해 결과를 가공하고 사용자 UI에 표현하기 적합한 XML 데이터 형태로 결과를 변환하는 작업을 수행한다.

3.2 추론 시스템 개선

추론 시스템 개선을 위해 데이터 구조 및 테이블 구조, EMPV의 도입, 추론 서비스와 데이터 요청 및 전달에 따른 프로세스 개선하고 캐쉬 관리기의 도입, 저장 구조에 적합한 트리플 관리를 위해 관리기를 개발하였다.

3.2.1 데이터 구조 및 테이블 구조

URI 저장 시 네임스페이스에 대한 prefix 사용, object의 자료형에 적합한 트리플 테이블의 생성 및 저장, backward chaining의 도입, 인스턴스 별 테이블 분할 등의 개선을 통해 저장 공간의 축소, URI 비교 연산 시 속도 개선, 필요 없는 형 변환을 제거하였다.
■ URI 네임스페이스
기존 자원에 부여된 URI는 풀네임 형태로 "http://www.kisti.re.kr/isrl#SOU_SEJ000001"을 사용해 왔다. 네임스페이스와 Prefix간의 매핑 정보를 별도로 관리하고, "isrl#SOU_SEJ000001"와 같이 Prefix를 사용해서 저장하도록 개선하였다.
■ Typed literal
기존에는 트리플의 자료형을 모두 문자열로 처리하고, 비교 연산이 필요한 경우 형 변환을 통해 연산을 수행하였다. Predicate가 datatype property인 경우 온톨로지 스키마를 참조하여 Object의 자료형에 맞게 테이블을 생성하고, 트리플을 저장하는 방식으로 개선하였다.
■ 테이블 구조
기존에는 모든 클래스의 인스턴스를 단일 테이블로 관리하였고, 전방 추론(forward chaining) 방식으로 subClassOf 등의 관계를 미리 확장하여 적용하였다. 클래스 별 인스턴스가 별도의 테이블로 관리되도록 수정하고, 필요한 경우 후방 추론(backward chaining) 방식으로 subClassOf 등의 온톨로지 스키마를 추론에 적용하도록 개선하였다.

3.2.2 EMPV 도입

트리플 저장 구조에서는 모든 자원이 트리플 형태로 관리되기 때문에, 하나의 객체를 중심으로 다중 속성을 만족하는 객체를 찾거나 또는 특정 객체와 관계된 다중 속성값을 제시하는데 비효율적이다. 이를 해소하기 하나의 객체를 중심으로 다중 속성값을 표현하기 위한 EMPV를 생성하고 활용하였다. EMPV는 기존의 트리플 저장 구조 외에 추가로 생성되는 테이블로서 다음과 같은 조건의 속성들을 대상으로 자동 생성되며, 관리자는 관리자 UI를 사용해서 EMPV 관계를 편집할 수 있다.
■ cardinality가 1인 데이터 타입 property인 경우
■ transitive property 이며, transitive 속성에 참여하는 property들이 모두 cardinality가 1인 경우
■ 2개 이상의 property가 연결되며, property들이 모두 cardinality가 1인 경우

표 1. RDF 트리플

subject	predicate	object
"isrl#PER_0000001"	rdf:type	isrl#Person
"isrl#PER_0000001"	isrl#nameOfPerson	"홍길동"
"isrl#PER_0000001"	isrl#emailAddressOfPerson	"hong@kisti.re.kr"
"isrl#PER_0000001"	isrl#hasInstitutionOfPerson	"isrl#INS_00000001"
"isrl#INS_00000001"	rdf:type	isrl#Institution
"isrl#INS_00000001"	isrl#nameOfInstitution	"KISTI"

표 2. EMPV 테이블

isrl#Person	"isrl#PER_0000001"
isrl#nameOfPerson	"홍길동"
isrl#emailAddressOfPerson	"hong@kisti.re.kr"
isrl#hasInstitutionOfPerson	"isrl#INS_00000001"

isrl#hasInstitutionOfPerso n_isrl#nameOfInstitution	“KISTI”

표 1과 표 2는 인력과 관련된 property들을 대상으로 EMPV가 설정된 예를 보여준다. 표 1에서는 인력 URI “isrl#PER_0000001”와 관계를 가지고 있는 property들의 예를 보여주며, 표 2에서는 표 1에 나타난 인력 URI와 관계를 가지는 property들이 모두 cardinality가 1인 경우에 생성될 수 있는 인력 URI에 대한 EMPV 테이블을 보여준다. 2개 이상의 property들이 EMPV에 반영될 경우는 property_property를 사용해서 표현하며, inverse 관계의 속성이 표현될 경우는 inv_property 형태로 표현한다. 이렇게 표현된 EMPV 테이블을 사용하는 경우 인력 URI와 관련된 다양한 property들을 한번에 접근할 수 있는 장점을 가지게 된다. EMPV 테이블은 기존의 property 테이블 외에 추가적으로 생성된 후 관리되며, 온톨로지 스키마가 변경된 경우에는 EMPV가 재 작성되어야 한다.

3.2.3 프로세스 개선

기존에는 UI에 보여주기 위한 정보를 추론시스템에서 한번에 반환하기 위해서 많은 수의 트리플을 조회하였고 또 그 값을 XML로 변환하고 UI에서 처리하는데도 많은 시간이 소요되었다. 또한 같은 추론서비스가 요청되는 경우에도 추론시스템이 반복적으로 같은 작업을 수행하였기 때문에 시스템의 성능이 저하되었다. 이를 개선하기 위해서 캐쉬 관리기를 도입하였고, 사용자 UI와 주고 받는 정보의 양을 조정하기 위해 추론 프로세스를 개선하였다. 또한 데이터 일관성을 위해서 단위 작업을 지정하여 트랜잭션을 관리하도록 하였다.

■ 캐쉬 관리기

추론 시스템에 대한 요청과 결과를 캐슁하여, 같은 요청에 대해 추론 과정을 거치지 않고 결과를 반환하는 기능을 수행한다. 인스턴스가 추가된 시간, 규칙이 변경된 시간, 온톨로지 스키마가 변경된 시간 등을 고려하여 캐쉬의 유효성 여부 관리한다.

■ 프로세스 개선

사용자 UI에 제공하기 위한 정보 단위를 페이지, 객체별로 구분하고, 추론 시스템과 사용자 UI는 Ajax 기술을 사용해서 필요한 정보만을 요청하고 반환하도록 하였다. 또한 단위 정보의 삽입 및 갱신에 따른 트랜잭션 단위를 설정하여 트리플과 EMPV 사이의 데이터 일관성을 보장할 수 있도록 하였다.

3.2.4 관리기 제공

관리자가 추론 시스템에 저장된 트리플이나 규칙, EMPV의 관리를 용이하게 할 수 있도록 관리자 UI를 제공하고 필요한 기능은 웹을 통해 수행할 수 있도록 하였다.

■ 트리플 관리기 개발

특정 URI 관련 자료의 추적 및 삭제를 위한 API 제공하고, UI를 통해 특정 URI와 관련된 트리플 리스트는 물론 트리플의 생성 시간, 유형에 따른 관리를 지원한다. 또한 트리플 저장소에 적재된 데이터 현황 모니터링 기능 제공한다.

■ 규칙 관리기

사용자가 온톨로지 스키마에 사용된 predicate와 기 확장된 expanded predicate를 사용하여 정확한 규칙을 생성하고, 적용할 수 있도록 관리자 UI를 제공한다. 또한 규칙을 적용해서 생성된 트리플을 확인할 수 있는 기능도 제공한다.

■ EMPV 관리기

관리자 UI를 통해서 EMPV 테이블에 반영될 수 있는 항목들을 클래스 중심으로 제시하며, 관리자는 UI를 통해서 EMPV에 predicate를 추가/삭제할 수 있다.

4. 추론 프로세스

추론시스템은 데이터 등록 프로세스와 데이터 요청 프로세스로 구분할 수 있다. OntoFrame®에서는 대용량 인스턴스의 효율적인 관리를 위해서 온톨로지 스키마와 인스턴스를 구분하고, 별도의 프로세스를 통해서 관리한다. 추론시스템에 등록되는 데이터는 온톨로지 스키마, 인스턴스 트리플, 규칙 및 EMPV 정보로써, 등록되는 데이터 유형에 따라 등록 프로세스는 다음과 같이 세가지 작업으로 구분된다.

■ 온톨로지 스키마 등록 프로세스

온톨로지 스키마에 해당하는 클래스, property, property 속성(cardinality, domain, range, …) 정보를 대상으로 스키마 유효성을 검증하고, 검증된 온톨로지 스키마를 트리플 형태로 표현하여 DBMS에 저장한다.

■ 인스턴스 등록 프로세스

인스턴스 정보를 트리플로 표현하고 추론 서비스 지원을 위한 추가 정보를 전처리 과정을 통해 트리플에 반영한 후 규칙을 적용하여 지식을 확장하고, EMPV 관계를 설정하여 DBMS에 저장한다.

■ 트리플, 규칙, EMPV 관리 프로세스

트리플 관리에 필요한 정보, 규칙의 생성 및 관리, EMPV의 생성 및 관리를 위한 정보를 DBMS에 저장한다.

그림 2는 온톨로지 스키마 등록 프로세스, 인스턴스 등록 프로세스, 관리 프로세스를 구분하여 보여준다. 온톨로지 스키마는 1~5의 단계를 거쳐 DBMS에 반영되며, 인스턴스는 A~K의 단계를 거쳐 DBMS에 반영된다. 관리자는 관리자 UI를 통해 트리플 관리, 규칙 관리, EMPV 관리 작업을 수행할 수 있다.

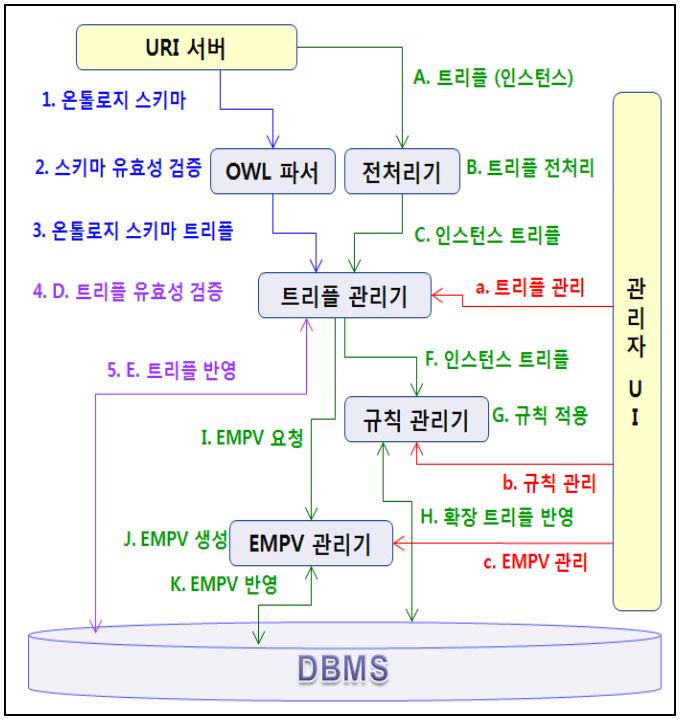

그림 2. 데이터 등록 프로세스

온톨로지 스키마는 온톨로지 스키마의 변경이 없는 경우 최초 1회에 한하여 등록 과정을 거치게 된다. URI 서버에서 전달 받은 온톨로지 스키마는 OWL 파서를 통해 스키마 유효성을 검증하고, 검증된 온톨로지 스키마는 트리플 형태로 트리플 관리기에 전달되어 DBMS에 등록된다. 이렇게 등록된 온톨로지 스키마 정보는 테이블 생성 및 트리플 저장, EMPV 관리, 추론 과정에서 SPAQRL을 SQL로 변환하는 작업 등에 지속적으로 활용된다. URI 서버에서 전달 받은 인스턴스 트리플은 전처리기를 통해서 추론 서비스를 지원하기 위한 정보를 트리플에 반영한 후, 트리플 형태로 트리플 관리기에 전달된다. 트리플 관리기에서는 온톨로지 스키마 정보를 이용해서 object의 자료형에 따라 트리플 정보의 형 변환을 수행하고 인스턴스와 predicate에 따라 각각의 테이블에 저장한다. 관리자에 의해 정의된 규칙을 적용하여 트리플을 확장하고 이를 테이블에 반영한다. EMPV 관리기는 온톨로지 스키마에 기반하여 자동으로 EMPV 정보를 생성하고, 관리자에 의해 확인 과정을 거친 후에, DBMS에 저장된 트리플을 대상으로 객체 별 다중 속성을 EMPV에 반영한다.

그림 3은 사용자 UI를 통한 추론 서비스 요청에 대해 결과를 생성하는 추론 시스템의 프로세스를 보여주고 있다. 추론 시스템은 시작과 동시에 DBMS에서 온톨로지 스키마 정보와 EMPV 정보를 로딩하여 활용한다. 사용자의 추론 서비스 요청은 SPAQRL로 표현되어 질의 처리기에 전달되며, 질의 처리기는 온톨로지 스키마 정보와 EMPV 정보를 사용해서 SPARQL 질의를 SQL 질의로 변환한다. 변환된 질의는 캐쉬 관리기를 통해 캐쉬 여부를 확인하고 캐쉬의 결과를 활용할 수 있으면 더 이상 단계를 진행하지 않고 결과를 반환하게 된다. 캐쉬에 존재하지 않는 경우 트리플 관리기와 EMPV 관리기를 통해 추론 결과를 생성한 후 후처리기를 통해 추론 서비스에 적합한 형태로 변환하는 작업을 수행하고, 이를 캐쉬에 반영한 후 추론 서비스 결과를 반환하게 된다.

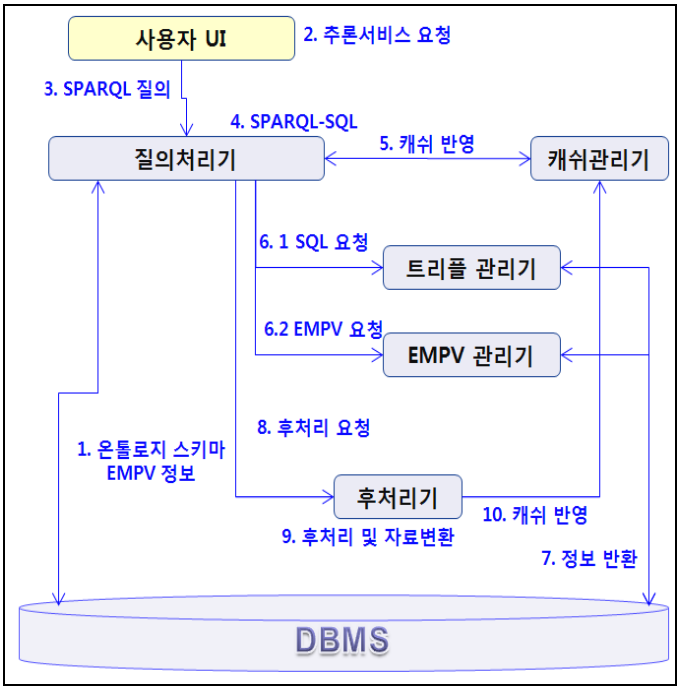

그림 3. 데이터 요청 프로세스

5. 결론

대량의 인스턴스를 대상으로 자원을 트리플로 표현하고 추론 서비스에 적합하도록 추론 시스템을 개선하였다. 이를 위해 네임스페이스 매핑과 자료형에 적합한 테이블의 생성 및 관리, 전방 추론과 후방 추론 데이터 및 온톨로지 스키마를 통한 전방 추론과 후방 추론을 활용할 수 있도록 테이블 구조를 개선하였다. 트리플 저장 구조의 자료 접근에 따른 비효율성을 회피하기 위해서 EMPV(Expanded Multiple Properties View) 도입하여 객체를 중심으로 다중 속성을 관리하고, 추론 프로세스 개선을 통해 추론 성능을 개선하였다. 규칙 및 트리플, EMPV 관리기의 도입을 통해 사용자 편의성을 높이고, 추론 서비스에 적합하도록 저장 구조를 관리하는 기능을 제공하였다.

향후 연구로는 개선된 시스템의 사용하여 시스템의 성능 평가를 통한 시스템 프로세스의 최적화, EMPV 테이블의 최적화 및 시스템 안정화를 고려하고, 대용량 정보를 대상으로 빠른 추론을 지원하기 위해 메모리 기반 데이터 처리 방식과 별도의 인덱싱 기법을 적용한 자료 접근 방식을 도입하려고 하고 있다. 또한 OWL, SPARQL의 확대 적용은 물론 SWRL을 사용한 규칙

정의 지원 확대 등을 통해 추론 시스템의 범용화 및 표준화를 지향하려고 한다. 또한 실제 데이터를 대상으로 저장 구조의 개선에 따른 오버 헤드 측정을 통해 저장 구조의 효율성을 확인하고, 추론에 소요되는 시간 측정을 통해 시스템 개선에 따른 항목별 유용성을 평가하고 개선하고자 한다.

[참고문헌]

[1] 이승우, 김평, 정한민, 강인수, 성원경. OntoThink-K 추론 서비스의 활용, KOSTI 2006, pp.377-383, 2006.

[2] 김평, 이승우, 이미경, 구남앙, 강인수, 정한민, 성원경. OntoStore-K: URI 기반 성과 관리 시스템, 한국정보과학회 제 33 회 추계학술발표회, 33 권, 2(B), pp.209-212, 2006.

[3] 강인수, 정한민, 이승우, 김평, 성원경. 국가과학기술 R&D 기반정보 온톨로지와 추론 모델링, 2006 한국컴퓨터종합학술대회, 2006.

[4] W3C. SPARQL Query Language, http://www.w3.org/TR/rdf-sparql-query/.

[5] 이미경, 김평, 정한민, 성원경. 지식기반 정보유통플랫폼(OntoFrame-K)의 추론 시스템, KOSTI 2006, pp. 84-90, 2006.

[6] Y. Guo, Z. Pan, J. Heflin. LUBM: A Benchmark for OWL Knowledge Base Systems, Journal of Web Semantics 3(2), pp.158-182, 2005.

[7] J. Broekstra, A. Kampman, F. van Harmelen. Sesame: A generic architecture for storing and querying rdf and rdf schema, ISWC2002, LNCS 2342, pp. 54-68, 2002.

[8] Z. Pan, J. Heflin. DLDB: Extending Relational Databases to Support Semantic Web Queries, In Workshop on Practical and Scalable Semantic Systems, 2003.

[9] J.Mei, L. Ma, Y. Pan, Ontology Query Answering on Databases, ISWC 2006, LNCS 4273, pp.445-458, 2006.

[10] E. Prud'hommeaux, A. Seaborne ed. SPARQL Query Language for RDF, W3C Working Draft 19 April 2005.

[11] K. Wilkinson, C. Sayers, H. Kuno, D. Reynolds. Efficient RDF Storage and Retrieval in Jena2. Proc. of the 1st International Workshop on Semantic Web and Databases, pp.131-151, 2003.

[12] E. Chong, S. Das, G. Eadon, J. Srinivasan. An Efficient SQL-based RDF Querying Scheme, VLDB 2005, pp. 1216-1227, 2005.