동일 개체를 위한 owl:sameAs 관리 서버

김평O, 이승우, 서동민, 정한민
한국과학기술정보연구원
{pyung, swlee, dmseo, jhm}@kisti.re.kr

owl:sameAs Synchronization Server for Same Objects

Pyung KimO, Seungwoo Lee, Dongmin Seo, Hanmin Jung
Korea Institute of Science and Technology Information(KISTI)

요 약

시맨틱 웹은 웹 환경에서 데이터의 의미를 표준화된 방법으로 표현함으로써 데이터의 상호 운용성을 보장하고 기계가 활용 가능한 데이터의 웹을 가능하게 해준다. 온톨로지에서 데이터는 식별자(URI)를 사용해서 의미가 명확화되고, 표준 기술 방법(RDF)를 통해서 어플리케이션 간 데이터의 통합 및 재사용을 가능하게 해준다. 최근 미국과 유럽을 중심으로 링크드 데이터 프로젝트를 통해서 시맨틱 데이터들의 상호 연계가 활발하게 추진하고 있다. 그러나 다양한 출처들의 데이터를 연계하는 과정에서, 동일한 객체에 서로 다른 식별자가 할당된 경우 식별자를 통한 시맨틱 정보 연계에 문제가 발생할 수 있다. OWL에서는 동일 객체에 대한 2개 이상의 식별자가 부여된 경우 owl:sameAs를 이용해서 식별자들이 동일 객체를 가리키고 있음을 명시한다. 본 연구에서는 서로 다른 식별자를 가진 객체들이 owl:sameAs를 사용해서 동일 객체로 표현되었을 경우, 동일 객체에 부여된 식별자 정보를 효과적으로 관리하고, 이를 서비스에 활용하기 위한 관리 서버를 설계하였다. 관리 서버를 통해 동일 객체에 대한 식별자들의 체계적인 관리는 물론, 동일 객체를 찾기 위한 질의 횟수를 감소시켜서 서비스 소요시간을 줄일 수 있다.

1. 서 론

시맨틱 웹은 데이터의 의미를 명확화하기 위해서 식별자(URI: Uniform Resource Identifier)와 표준화된 지식 표현 기술(RDF: Resource Description Framework)을 사용하며, 데이터 공유 및 데이터 연계를 통한 부가 정보의 생성에 효율적인 기술이다[1]. 시맨틱 웹의 지식 표현 모델로 사용되는 온톨로지는 지식 개념의 의미를 표현하는 모델로서, OWL(Web Ontology Language)[2]이 W3의 표준으로 많이 사용되고 있다. 온톨로지의 모든 객체는 식별자를 기반으로 표현되며, 식별자는 데이터의 연계 및 병합에 아주 중요한 연결자 역할을 수행한다.

OWL에서는 서로 다른 식별자가 동일 개체를 표현하는 경우 owl:sameAs 속성을 이용해서 표현하고, 두 개체가 동일한 객체임을 명시하게 된다[3]. 서로 다른 식별자를 가지는 두 객체가 동일 객체로 판별된 경우, 두 객체는 모든 속성을 공유하게 되기 때문에 객체들의 속성 정보를 효과적으로 공유하고 관리하기 위한 연구[4]도 수행되었다. 하지만 2개 이상의 식별자를 가지는 동일 객체의 속성 정보를 처리하기 위해서는 전방추론(Forward chaining)을 통해 미리 모든 속성관계를 확장하거나 또는, 질의 처리 시점에서 동일 객체들에 부여된 식별자를 모두 검색한 후 해당 식별자를 사용해서 모든 속성을 검색해야 한다.

데이터의 출처와 생성 시점, 생성자에 따라 동일 객체에 대해 서로 다른 식별자가 부여될 수 있으며, 다양한 출처의 시맨틱 데이터가 연계되는 과정에서 동일 객체들에 부여된 서로 다른 식별자를 발견하고 이를 연계 및 활용하기 위한 연구가 점점 중요해지고 있다. 본 연구에서는 owl:sameAs 관계가 부여된 식별자들을 통합 관리하기 위한 계층적 관리 서버를 설계하고, 관리 서버를 통해 동일 객체에 부여된 식별자들을 한번에 얻는 것이 가능하다. 특히 동일 관리 환경에서 관리되는 다수의 분산 온톨로지들을 대상으로 owl:sameAs 관계가 존재하는 경우 계층적인 관리 서버를 통해 owl:sameAs 정보를 관리 서버 간 실시간 동기화함으로써 서버 별 동일 객체 정보를 얻기 위한 SPARQL 질의 횟수를 단축시켜 준다.

2장에서는 시맨틱 웹 환경에서 데이터를 연계하기 위한 연구, 동일 객체를 식별하고 식별자를 할당하기 위한 연구, owl:sameAs를 관리하기 위한 연구에 대해서 기술하고, 3장에서는 본 연구에서 제안하는 관리 서버의 구조와 프로세스에 대해서 기술하고, 4장에서는 결론 및 향후 연구 방향에 대해서 기술한다..

2. 관련연구

위키피디아에서는 링크드 데이터(Linked Data)를 “URI와 RDF를 이용해 시맨틱 웹 상에 널려있는 데이터,

정보, 지식을 노출하고 공유하며 연결하기 위해 추천되는 최고의 방법"라고 설명하고 있다[5]. 최근 미국과 유럽을 중심으로 진행중인 링크드 데이터 프로젝트[6]는 다양한 출처의 시맨틱 데이터를 공유하고 연계하기 위한 연구로서, 데이터 연계를 통한 부가 데이터 생성이라는 측면에서 관심을 받고 있다. 링크드 데이터 프로젝트에 포함된 시맨틱 데이터의 경우에도 데이터 출처 별로 동일 객체에 서로 다른 식별자를 할당하는 경우가 빈번하게 발생하고 있다. 링크드 데이터는 서로 다른 출처의 데이터를 연계 및 활용하는 과정에서 서로 다른 식별자가 부여된 동일 객체가 출판되기 쉬우며, 서로 다른 식별자를 가진 동일 객체에 대한 관리가 필요하다. 영국의 South Ampton 대학에서는 서로 다른 식별자를 가지는 동일 객체들을 연계하고 관리하기 위해서 번들(Bundle)을 사용해서 동일 객체들에게 부여된 식별자를 동일한 집합으로 관리하고 활용하기 위한 연구를 수행하였다[7]. 이 연구 결과를 소개하는 sameAs 사이트[8]에서 "Tim Berners Lee"를 검색하면 Sindice[9]검색 엔진을 이용해서 88개의 서로 다른 식별자가 검색되고, 10개의 서로 다른 객체를 의미하는 10개의 번들로 구분되어 나타나고 있음을 알 수 있다. 분산 온톨로지 환경에서 동일 객체에 대한 서로 다른 식별자가 많이 존재하기 때문에 식별자의 효율적인 관리가 필수적으로 요구된다.

동일 개체를 식별하기 위한 연구는 주로 문헌정보에 나타나는 저자를 중심으로 동일 저자의 연구 성과를 검색하기 위해서 사용되었다[10,11]. KISTI에서도 문헌정보와 검색 엔진을 이용한 저자 식별 연구를 통해 문헌에서 다양한 정보를 활용해서 식별된 동일 저자에게 동일한 식별자를 부여하기 위한 노력을 수행하였고[10], 대용량 데이터를 대상으로 저자 식별 실험을 할 수 있는 평가셋도 개발하였다[12]. 하지만 모든 동일 개체가 동일한 식별자를 부여받고 관리되는 것은 일반적인 상황에서 벗어나며, 분산 서버에서 서비스중인 온톨로지를 대상으로 동일 객체에 대한 서로 다른 식별자를 체계적으로 관리하고 서비스에 활용하기 위한 방법은 제시되지 못하였다.

3. sameAs 관리 서버

동일 객체에 대한 식별자가 서버 별로 분산되어 있는 경우 연관 속성들을 모두 찾기 위해서는 각 서버 별로 분산되어 있는 동일 개체 식별자를 찾은 후, 식별자를 모두 사용해서 서버 별 질의를 작성해야 하는데 이 과정은 객체들이 웹을 통해 연계되어 있기 때문에 쉽지 않다. 또한 또 다른 식별자를 사용하는 객체가 동일 객체로 판별된 경우 이를 효과적으로 관리하기 위한 방법도 요구된다.

본 연구에서는 분산 서버 환경에서 온톨로지들이 연계 서비스 되는 경우, 서로 다른 식별자를 가진 동일 개체들의 정보를 owl:sameAs로 표현하고 이를 관리 서버가 동기화하여 관리한다. 관리 서버는 서비스 시점에 동일 개체를 찾기 위한 SPARQL 질의 대신 메모리에서 동일 개체에 대한 식별자를 얻고 이를 이용한 SPARQL 질의 재작성을 통해 서비스 소요 시간을 줄이기 위해 활용된다. 본 장에서는 owl:sameAs 정보를 관리 서버의 계층 구조를 통해 효과적으로 관리하고 서비스에 활용하기 위한 관리 서버 구조와 프로세스에 대해서 기술한다.

3.1 관리 서버 구조

관리 서버는 분산 온톨로지 환경에서 판별된 동일 개체들에 대한 서로 다른 식별자를 효과적으로 공유하고 활용하기 위해서, 관리 서버를 계층적으로 설계하고, 동기화 규칙을 통해 owl:sameAs 관계를 관리한다.

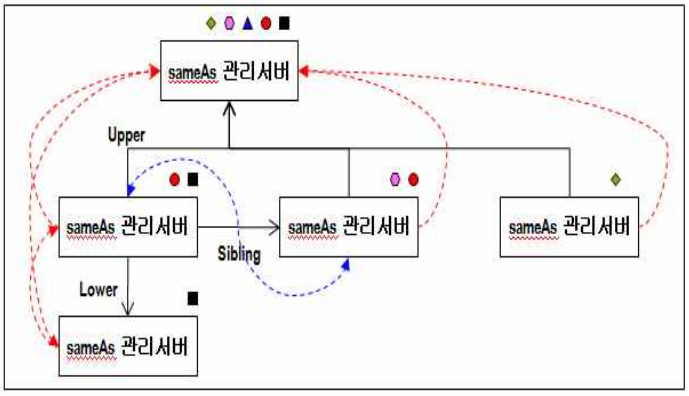

그림 1. 관리 서버의 계층 구조

관리 서버는 그림 1과 같이 계층 구조를 가지며, 관리 서버별 동기화 규칙에 따라 동일 객체에 대한 식별자 관계를 동기화하는 과정을 수행하게 된다. 빨간색 점선 화살표는 해당 관리 서버의 정보가 상위(Upper) 관리 서버에 포함되는 방향을, 파란색 점선 화살표는 형제(Sibling) 관리 서버에 동기화 되는 방향을 보여준다. 계층 구조상에 상위 관리 서버는 하위 관리 서버의 모든 동일 개체의 식별자 관계를 보관하고 있으며, 상하위 관계는 이행성 관계(transitive relation)를 가진다. 최상위 관리 서버는 모든 하위 관리 서버들의 모든 동일 개체의 식별자 정보를 가지게 된다. 관리 서버의 정보가 상하위 관계와 형제 관계에 식별 정보의 포함 여부는 그림 1의 작은 도형의 포함 관계를 통해 나타난다.

관리 서버의 계층 구조는 그림 2와 같은 서버별 동기화 규칙 정보를 통해 설정된다. 두 개의 관리 서버 A, B가 존재하는 경우 A가 B를 형제 관계로 설정하고, B도 A를 형제 관계로 설정한 경우 두 개의 서버는 형제 관계로 동일 개체 정보를 동기화하는 과정을 수행한다.

동일 개체에 대한 식별자 정보는 sameAs 부분에서 기술되며, 해당 관리 서버에서 추가한 동일 객체 정보들은 "sameAs(OWN)"로 기술된다. sameAs 출처별로 두 개의 식별자가 쌍으로 존재하며 이 두 식별자는 동일 개체를 의미한다.

```
#Name
  http://isrl.kisti.re.kr
#Sibling (URL)
  http://www.kats.go.kr/sameAs
  http://www.moj.go.kr/sameAs
#Upper
  http://www.kisti.re.kr/sameAs
#Lower
  http://isrl.kisti.re.kr/Sub1/sameAs
#Web Services(URL)
  set : setSameAsProperty (URI1, URI,cmd)
  get : getSameAsProperty (URI1)
#manager
  http://isrl.kisti.re.kr/PER_0001
#Date
  11/28/2009 00:00:00
#sameAs (OWN)
  http://isrl.kisti.re.kr/Per_001,http://isrl.kisti.re.kr/Per_002
#sameAs (http://www.kats.go.kr/sameAs )
  http://isrl.kisti.re.kr/Per_001,http://isrl.kisti.re.kr/Per_005
```

그림 2. 동기화 규칙

3.2 관리 서버 프로세스

관리 서버는 기본적으로 그림 2와 같이 동일 객체 관계의 식별자 쌍을 추가/삭제하기 위한 API인 "setSameAsProperty"와 식별자를 사용해서 동일 객체를 가리키는 다른 식별자들을 얻기 위한 API인 "getSameAsProperty"를 제공한다.

3.2.1 서버 동기화

새로운 동일 객체 식별자 쌍을 추가하거나 기존 식별자 쌍을 제거하는 작업은 웹 서비스를 통해 관리 서버에서 제공하는 "setSameAsProperty" 함수를 호출하게 된다. 이 작업은 주로 관리자에 의해 수행되며, 관리자가 동일 개체 식별자 쌍을 추가하거나 삭제하는 경우 그림 3과 같이 해당 관리 서버의 상위 관계와 형제 관계에 있는 관리 서버와 동일 객체 식별자 쌍 정보를 동기화한다.

이렇게 관리 서버의 동일 객체 식별자 쌍을 동기화하는 작업은 관리 서버의 모든 상위 관계에 있는 관리 서버와 형제 관계에 있는 모든 관리 서버를 대상으로 수행된다.

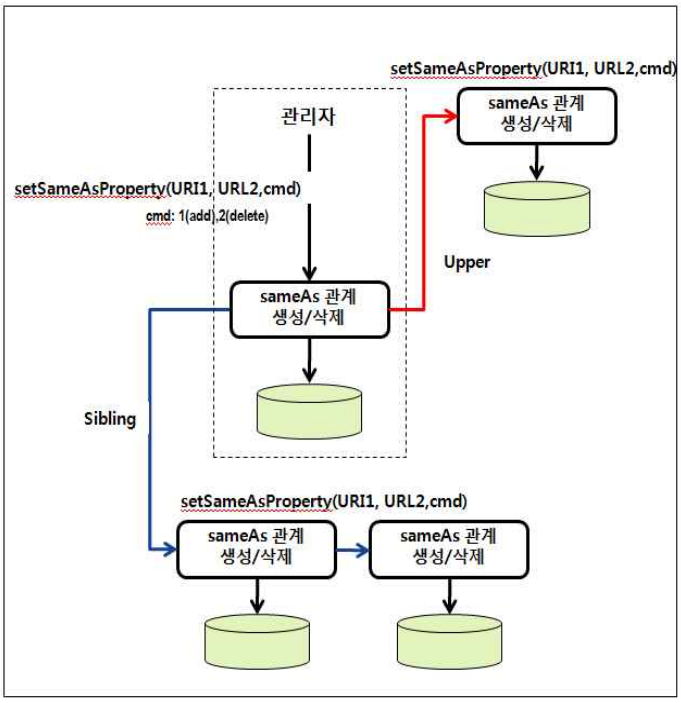

그림 3. 동기화 프로세스

3.2.2 질의 처리

owl:sameAs 정보가 별도로 관리되지 않는 경우 분산 온톨로지를 대상으로 SPARQL 질의를 처리하는 경우 그림 4와 같이 SPARQL 질의를 통해 각 서버 별로 동일 객체에 대한 식별자를 얻기 위한 질의를 수행한다.

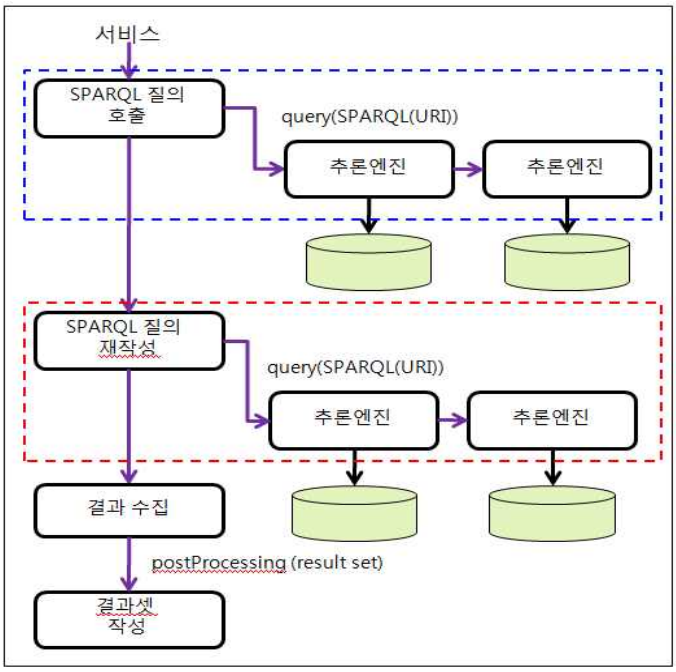

그림 4. 일반 검색 프로세스

이렇게 수행된 질의를 통해 얻은 동일 객체에 대한 식별자 정보를 이용해서 SPARQL 질의를 재작성해서 서비스에 필요한 연관 속성을 얻고 결과를 생성하는 프로세스를 거치게 된다. 추론엔진이나 트리플 스토어를 통한 SPARQL 질의는 처리에 소요되는 시간과, 다수의 분산 온톨로지를 대상으로 질의를 처리해야 하기 때문에 전체적인 서비스 응답 속도가 늦어지게 된다.

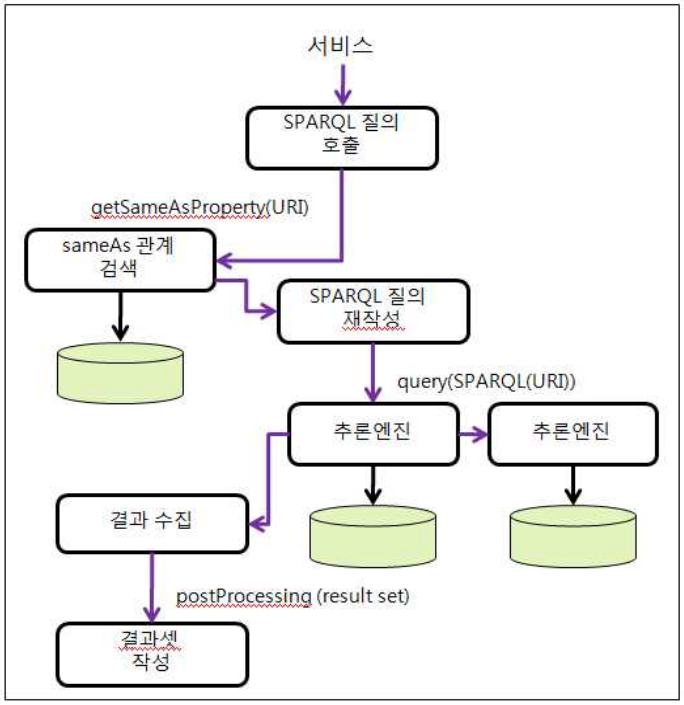

그림 5. 관리 서버를 이용한 검색 프로세스

그림 5의 경우 관리 서버를 통해 해당 서비스에 관여하고 있는 분산 온톨로지의 동일 객체에 대한 식별자 정보를 한 번에 획득하고, 이를 사용해서 SPARQL 질의를 재작성해서 분산 서버에 한 번의 SPARQL 질의를 수행함으로써 최종 결과를 생산하는데 필요한 정보를 얻을 수 있다.

특히 서비스에 관여하는 분산 서버의 수가 많고, 동일 객체에 대한 다수의 서로 다른 식별자가 존재하는 경우 서버 별 SPARQL 질의 횟수를 감소시킴으로써 서비스 응답 속도를 개선할 수 있다. 또한 owl:sameAs 관리 서버는 관리 서버가 없는 다른 서버와의 연계가 용이하며, 다수의 분산 서버에 존재하는 동일 객체에 대한 식별자를 효율적으로 통합 관리할 수 있는 환경을 제공한다.

4. 결 론

시맨틱 웹을 통해서 다수의 온톨로지가 연계 및 활용되면서 객체들의 식별자에 대한 할당 및 관리 방법에 대한 중요성이 더욱 더 부각되고 있다. 식별자는 데이터 연계에 중요한 역할을 수행하고 있으며, 분산 온톨로지 환경에서 동일 객체들의 식별자를 효율적으로 관리하고 서비스할 수 있는 방법이 요구된다.

본 연구에서는 owl:sameAs 정보를 관리하기 위한 계층적 구조의 관리 서버를 설계하고, 서버 간 동일 객체에 대한 식별자 정보 동기화를 통해 다수의 분산 온톨로지 환경에서 서비스 질의 시점에 활용하기 위한 방법을 제시하였다. 기존 서비스 환경에서 owl:sameAs 관계를 획득하기 위해서 요구되었던 SPARQL 질의를 대체하기 위한 방법으로서, 관리 서버는 동일 객체에 대한 식별자 정보를 계층 구조상 상위 관리 서버와 형제 관리 서버와 동기화 해서 관리하고, 이를 서비스에 바로 활용함으로써 일반적인 검색 프로세스를 보다 효율적으로 재구성할 수 있다. 또한 관리 서버를 통해 동일 객체에 대한 식별자 정보를 통합 관리 할 수 있는 환경을 제공하였다. 향후에는 관리 서버가 존재하지 않는 일반 서버와 관리 서버의 정보를 연계하고 실시간 동기화 과정에서 발생하는 다양한 장애를 해소하기 위한 방법에 대해서 연구하겠다,

참고문헌

[1] J. Hendler, “Agents and the Semantic Web”, IEEE Intelligent Systems, Volume 16, Issue 2. pp 30-37, 2001.
[2]http://www.w3.org/TR/owl-features
[3]http://www.w3.org/TR/owl-ref/#sameAs-def
[4]강인수 외 5, “시맨틱 웹 온톨로지에서의 OWL sameAs 적용”, 정보과학회 논문지, 34권 4호, pp 359-367.
[5] http://en.wikipedia.org/wiki/Linked_data
[6] http://linkeddata.org
[7] Hugh Glaser 외 6, “Research on Linked Data and Co-reference Resolution”, International Conference on Dublin Core and Metadata Applications 2009.
[8]http://sameas.org
[9] http://sindice.com
[10] 강인수 외 6, “On co-authorship for author disambiguation”, In journal of Information processing & management, 45(1), pp 84-97, 2009.
[11]D. A. Pereira 외 5, "Using web information for author name disambiguation", In Proceedings of ACM/IEEE-CS Joint Conference on Digital Libraries(JCDL), pp.49-58, 2009(6).
[12]강인수 외 4, “저자 식별을 위한 대용량 평가셋 구축”, 한국콘텐츠학회 논문지, 9월 11호, pp 455-464.