

온톨로지 통합을 위한 변환 규칙의 확장

김선경^{*}, 김병곤¹, 이재호², 임해철

홍익대학교 컴퓨터공학과, ¹부천대학 e-비즈니스과, ²경인교육대학교 컴퓨터교육과
 {skkim^{*}, lim}@cs.hongik.ac.kr, bgkim@bc.ac.kr, jhlee@gin.ac.kr

The extensible mapping rules for ontology integration

Sun Kyung Kim^{*}, Byung Gon Kim¹, Jaeho Lee², Hae Chull Lim

Dept. of Computer Eng., Hong Ik Univ., ¹Dept. of e-Business, Bucheon College,

²Dept. of Computer Education, Gyungin National University of Education

요 약

웹에 대한 사용자의 다양한 요구와 더불어 시맨틱 웹에 대한 연구가 활발하게 이루어지고 있다. 이러한 연구 중에 대표적인 예가 바로 온톨로지의 구축에 대한 연구이다. ICS-FORTH는 온톨로지 구축을 위한 대표적인 시스템으로 SWIM을 이용한 분산된 환경에서의 독립된 온톨로지 즉, 관계형 데이터베이스와 XML로 된 온톨로지의 통합이 가능하도록 하였다. 또한, 통합된 온톨로지를 RDF/S를 기반으로 하는 뷰의 형태로 사용자에게 제공하여 추론이 가능한 질의 처리가 가능하도록 하였다. 이에 따라, 본 논문에서는 W3C에서 제안한 OWL을 통합의 대상으로 통합의 범위를 확장하기 위한 기법을 제안한다. 제안한 기법은 RDF/S를 확장한 다양한 관계의 기술이 가능하기 때문에 사용자가 RQL 형태의 질의를 미들웨어에서 처리할 수 있는 형태로의 변환하는 기존 기법의 확장된 형태이다. 이러한 확장을 통하여 기존 온톨로지 통합의 범위를 확장하여 보다 많은 사용자의 다양한 요구에 대응할 수 있다.

1. 서 론

웹에 대한 사용자의 다양한 요구와 더불어 시맨틱 웹에 대한 관심이 증대되면서 W3C에서는 온톨로지를 기술하기 위한 언어로 OWL[1]을 제안하였다. OWL에 대한 초기기의 연구는 주로 OWL 자체와 향후 연구 방향을 제시하고 있는 연구[2]가 주류를 이루었고, 최근에는 온톨로지 구축, 유지 및 통합하기 위한 연구에 대한 관심이 증대되고 있다. 특히, 온톨로지 저장 시스템인 ICS-FORTH에서는 SWIM을 이용하여 독립된 온톨로지의 의미적인 통합이 가능하도록 하였다. 즉, 통합의 대상물 실제 세계에서 일반적으로 많이 사용되고 있는 관계형 데이터베이스와 시맨틱 웹의 근간을 이루고 있는 XML등의 저장소를 통합의 대상으로 하여 사용자에게 RDF/S로 된 뷰의 형태를 제공함으로써 추론이 가능하도록 하였다. 이에 따라, 본 연구에서는 온톨로지 통합의 대상물 OWL로 확장하기 위하여 기존 뷰에서 표현하고 있는 6개의 릴레이션 이외에 OWL에서 기술하고 있는 다양한 형태의 릴레이션을 표현하기 위한 9개의 릴레이션을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 SWIM과 확장의 대상이 되는 OWL에 대하여 설명한다. 3장에서는 OWL을 표현하기 위한 릴레이션과 질의 유형에 대하여 설명하고, 4장에서는 결론과 향후 연구를 설명한다.

경에서 여러 온톨로지를 대상으로 질의가 가능하다.

ICS-FORTH 시스템은 온톨로지를 RDB, XML 저장소를 이용하여 저장하고, 각각의 온톨로지는 RDF/S 기반의 뷰로 구성되어 사용자에게 제공된다. 사용자는 각자 가상의 뷰에 대하여 RQL[4] 형태로 질의한다. 하지만, 내부적으로 SWIM에서는 물리적 저장 위치에 상관없이 전체 온톨로지에 대한 스키마 정보를 유지하고, RVL[5]로 기술된 가상의 뷰에 질의하기 위하여 RQL을 뷰에 대한 질의 형태로 변환하여야만 한다. 또한, 뷰에 대한 질의 후 실제 인스턴스 값을 반환하기 위하여 RVL로 기술된 질의어를 RDB, XML 저장소에서 사용가능한 질의형태(예, SQL, XPath)로 변환 후 실제 인스턴스 값을 반환한다. 이때, 독립된 온톨로지들을 하나의 뷰로 만들기 위한 과정에서 가장 중요한 쟁점중의 하나는 하위 온톨로지를 의미적인 손실 없이 하나의 가상 뷰로 어떻게 표현하느냐하는 점이다. 이러한 관점에서 기존 ICS-FORTH에서는 5개의 릴레이션을 이용하여 RDF/S를 의미적인 손실 없이 표현할 수 있었지만, 다른 온톨로지 언어에 대한 표현 방법의 다양성 측면에서 온톨로지 구성요소 간의 관계에 대한 확장의 필요성이 대두되고 있다. 다음은 [3]에서 제안된 RDF/S 기술하기 위한 릴레이션을 나타낸다.

표 1. RDF/S를 기술하기 위한 릴레이션

릴레이션 이름	설명
C_EXT(c,x)	• 클래스 C의 인스턴스는 x
C_SUB(c,d)	• 클래스 C는 클래스 D의 서브 클래스
PROP(c,p,d)	• 속성 p의 도메인은 c이고, 레인지는 d
P_EXT(x,q,y)	• 리소스 x와 y는 속성 q라는 관계를 가짐
P_SUB(p,q)	• 속성 p는 속성 q의 서브프로퍼티

2. 관련 연구

2.1 SWIM(Semantic Web Integration Module)

ICS-FORTH 시스템은 시맨틱 웹의 다양한 연구 분야 중 독립된 온톨로지들을 통합 및 저장을 위한 대표적인 미들웨어 시스템이다[3]. 사용자는 RDF/S 형태로 된 뷰를 제공받음으로써 이질적인 한

본 연구는 한국과학재단 지호과학연구사업 (과제번호 : R01-2004-000-10586-0(2005))의 지원을 받아서 작성되었음

2.2. OWL

온톨로지를 기술하기 위하여 제안된 언어들 중에서 최근 W3C에서 제안된 대표적인 언어로서, 기존 RDF/S를 이용한 온톨로지의 표현을 확장 및 강화하기 위하여 나타났다[1]. 또한, 표현이 대상이 되는 클래스, 속성, 인스턴스 간의 복잡한 관계(예, inverse, union, intersection, ...)에 대한 기술이 가능하기 때문에 RDF/S를 이용한 질의 유형 보다 확장된 형태의 다양한 질의 처리가 가능하다. 즉, 온톨로지의 다양한 표현에 대한 보다 많은 제약사항을 기술하기 때문에 클래스와 구성원간의 관계를 집합이란 개념을 통하여 온톨로지로는 정의되지 않는 사실들에 대한 논리적인 추론이 가능하기 때문에 최근 이를 이용한 다양한 연구가 진행되고 있다[2]. 따라서, 시맨틱 웹에서의 위치는 더욱 중요해지고 있고, 온톨로지 저장 시스템에서는 OWL을 위한 저장 구조는 반드시 고려되어야만 한다.

3. OWL을 고려한 릴레이션의 표현

본 논문에서는 기존 ICS-FORTH에서의 통합된 온톨로지의 구축의 대상이 되는 RDB, XML 이외에 OWL을 통합의 대상으로 한다. 따라서, OWL의 다양한 관계를 하나의 뷰로 표현하기 위하여 기존에 사용되었던 5개의 릴레이션 외에 9개의 릴레이션을 제안한다. 제안한 9개의 릴레이션은 OWL-DL로 기술된 온톨로지를 SWIM에서 사용할 수 있는 형태로 클래스, 속성, 인스턴스등의 다양한 관계의 기술이 가능하다. <표 2>는 본 논문에서 제안하는 OWL을 표현하기 위한 릴레이션을 나타낸다.

표 2. OWL을 기술하기 위한 릴레이션

릴레이션 이름	설명
E_CLA(c, d)	• 클래스 C와 D는 동치관계
E_PRO(p, q)	• 속성 p와 q는 동치관계 (같은 인스턴스를 가져야 함)
D_EXT(x, y)	• 인스턴스 x와 y는 서로 다름
A_EXT(x ₁ , ..., x _{n-1})	• 인스턴스 x ₁ , ..., x _{n-1} 은 서로 다른 관계
I_CLA(c, d, e)	• 클래스 C는 클래스 D, E에 각각 속함
U_CLA(c, d, e)	• 클래스 C는 클래스 D, E의 합집합
C_CLA(c, d)	• 클래스 C는 클래스 D에 속하지 않는 모든 인스턴스를 멤버로 함
O_EXT(c, x ₁ , ..., x _{n-1})	• 클래스 C는 인스턴스 x ₁ , ..., x _{n-1} 을 멤버로 가짐
D_CLA(c, d)	• 클래스 C와 D는 다른 구성원으로 구성됨

3.1 릴레이션(Relations)

· E_CLA(c, d)

두 클래스는 반드시 동일한 인스턴스를 가져야만 하는 관계를 표현하는 릴레이션으로 가령, 실제 인스턴스는 한 문서에만 존재하고 다른 문서에는 클래스의 정의와 전자의 클래스와 동치관계만을 표현하는 클래스가 있다면, 후자의 클래스에 질의 할 경우에도 전자의 클래스에 접근이 가능하기 때문에 질의처리를 좀 더 유연하게 할 수 있다.

· E_PRO(p, q)

스키마 레벨에서 프로퍼티와 프로퍼티간의 동치관계를 기술하기 위한 릴레이션이다. 위의 E_CLA(c, d)의 경우와 같이 같은 인스턴스를 가지는 프로퍼티간의 관계를 기술하기 위한 것이다.

· D_EXT(x, y)

인스턴스 간의 관계를 기술하기 위한 릴레이션이다. RDF/S에서 같은 클래스는 같은 인스턴스를 갖는다. 하지만, 동일한 이름의 클래스가 다른 인스턴스를 갖는 즉, 동음이의어와 같은 관계일 경우, 두 클래스간의 인스턴스를 구분해주어야만 하는 경우, 인스턴스간의 관계에 대한 정의가 가능하다.

· A_EXT(x₁, ..., x_{n-1})

D_EXT(x,y)의 확장 형태라고 할 수 있다. 위의 경우와 마찬가지로 RDF/S의 확장 형태로 클래스에 속해있는 다수의 인스턴스들간의 관계에 대한 기술을 위한 변환 규칙이다.

· I_CLA(c, d, e)

클래스간 교집합의 관계를 이용한 관계를 정의한다. 즉, 클래스간 포함관계를 이용하였기 때문에 질의 대상이 되는 클래스는 나머지 두 클래스가 가지고 있는 조건을 포함하는 관계 즉, 클래스 C는 클래스 D와 E를 포함하는 관계가 반드시 성립해야만 하기 때문에 클래스 간 관계를 이용한 조금 더 복잡한 형태의 질의가 가능하다.

· U_CLA(c, d, e)

OWL에서는 관계형 데이터베이스에서와 다르게 클래스들을 합집합의 관계가 되었을 때, 대상이 되는 클래스의 상위에 새로운 클래스를 정의할 수 있다. 즉, 사용자가 서로 다른 문서에 존재하는 클래스들을 하나의 문서와 같이 표현이 가능하다. 따라서, 다른 문서에 존재하는 클래스들을 합집합의 관계를 이용하여 표현할 경우, 여러 문서를 일일이 검색하지 않는 즉, 검색의 범위를 줄일 수 있다.

· C_CLA(c, d)

클래스 D의 여집합에 속하는 클래스를 기술하기 위한 것으로 사용자가 어떠한 것은 제외하고 라는 것을 클래스간의 관계로 표현할 때 유용하게 사용가능한 릴레이션이다.

· O_EXT(c, x₁, ..., x_{n-1})

클래스 C가 가질 수 있는 인스턴스들을 열거하기 위한 것으로 클래스와 인스턴스간의 관계는 반드시 직접적인 관계만을 허용한다. 즉, 클래스가 가질 수 있는 인스턴스들을 제한하였기 때문에 질의의 범위를 줄일 수 있다.

· D_CLA(c, d)

클래스와 클래스 간의 관계를 기술하기 위한 속성으로 클래스 D와 클래스 C는 동시에 같은 구성원을 가질 수 없다. 따라서, 본 릴레이션으로 선언된 클래스들 간의 관계와는 다르게 두 클래스간의 관계를 좀 더 명확히 기술할 수 있다.

3.2 질의 유형

다음은 본 논문에서 제안하는 릴레이션이 기술되어 있는 OWL 문서의 일부분이다. <그림 1>은 영화에 관계된 내용을 나타내는 OWL 문서이고, <그림 2>는 <그림 1>에 대한 스키마를 나타낸다. <그림 2>에서의 최상위 클래스인 'Movie', 'Director'와 속성 'madeby'은 같은 문서에 존재할 수도 있지만, 다른 문서에 존재할 수도 있다.

```

<owl:Class rdf:ID="Movie">
...
<owl:Class rdf:ID="Cinema">
<owl:EquivalentClass rdf:resource="#Movie"/>
</owl:Class>
<owl:Class rdf:ID="Genre">
<owl:UnionOf rdf:parseType="Collection">
<owl:Class rdf:ID="Romantic">
<rdf:subClassOf "#Genre"/>
</owl:Class>
<owl:Class rdf:ID="Thriller">
<rdf:subClassOf "#Genre"/>
</owl:Class>
</owl:UnionOf>
<owl:Class rdf:ID="Actor">
<owl:Class rdf:ID="Player">
<owl:IntersectionOf rdf:parseType="Collection">
<owl:Class rdf:ID="Male">
<owl:IntersectionOf
<rdf:subClassOf rdf:resource="#Players">
</owl:IntersectionOf>
<owl:Class rdf:ID="Actress">
<owl:IntersectionOf rdf:parseType="Collection">
<owl:Class rdf:ID="Female">
<owl:IntersectionOf
<rdf:subClassOf rdf:resource="#Players">
</owl:IntersectionOf>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasGenre">
<rdf:domain rdf:resource="#Cinema"/>
<rdf:range rdf:resource="#Genre">
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="casting">
<rdf:domain rdf:resource="#Cinema">
<rdf:range rdf:resource="#Players"/>
</owl:ObjectProperty>
<owl:ObjectProperty>
<rdf:domain rdf:resource="#Cinema"/>
<rdf:range rdf:resource="#Genre">
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="title">
<rdf:domain rdf:resource="#Cinema"/>
<rdf:domain rdf:resource="#Romantic"/>
<rdf:domain rdf:resource="#Thriller"/>
<rdf:range rdf:resource="#xsd:string"/>
</owl:DatatypeProperty>
...
    
```

그림 1. 영화 OWL 문서

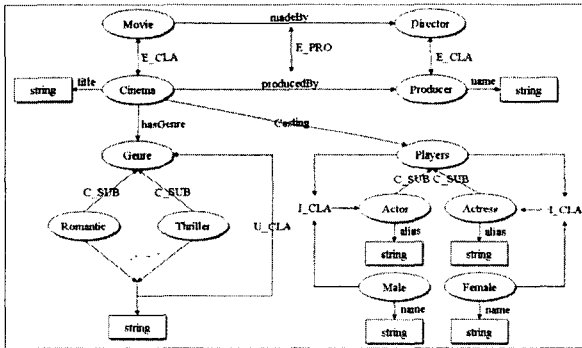


그림 2. 영화 OWL 문서의 스키마

다음은 <그림 1,2>를 통해 나타난 OWL 문서와 스키마 정보를 이용하여 사용자가 제시할 수 있는 다양한 형태의 질의 중 equivalent, intersection, union 등에 대한 질의 변환 규칙의 예를 나타낸 것이다.

유형 1.

RQL	{x:\$C}, {y:\$D}, x=y
Relations	C_EXT(c,x), C_EXT(d,y), E_CLA(c,d)

위의 RQL 질의 유형은 같은 인스턴스를 가져야만하는 E_CLA 관계에 있는 클래스의 관계를 이용한 질의 유형이다. <그림 2>에서 'Movie' 클래스는 'Cinema' 클래스와 E_CLA 관계이다. 즉, 'Movie' 클래스가 가질 수 있는 인스턴스는 'Cinema' 클래스가 가질 수 있는 인스턴스와 일치한다. 따라서, 사용자가 'Movie' 클래스를 대상으로 질의하였을 경우, 'Cinema' 클래스가 가질 수 있는 모든 인스턴스에 대한 접근이 가능하다. 이러한 이유로, 'S.F 영화의 종류를 모두 나열하라'란 질의가 있을 경우, 사용자는 'Cinema' 클래스에 대한 자세한 정보를 알지 못한다고 해도 'S.F' 클래스의 인스턴스에 대한 결과값의 반환이 가능하다.

유형 2.

RQL	{y, !({x:\$C} ⋈ {z:\$E})}
Relations	I_CLA(c, d, e), C_EXT(c, x), C_EXT(d, y), C_EXT(e, z)

위의 질의 유형은 두 개의 질의를 조인하여 질의 결과를 반환해야 하는 I_CLA 관계에 있는 클래스에 대한 질의 유형이다. <그림 2>에서 'Players' 클래스와 'Male' 클래스 모두 포함하는 클래스가 'Actor' 클래스이다. 즉, 'Actor' 클래스는 'Player' 이기도 해야 하고 'Male' 이어야 한다. 따라서, "남자들 중에서 배우인 사람의 이름을 검색하라"와 같은 질의인 경우, 기존 RQL에서와 같이 남자들의 이름과 배우의 이름을 모두 검색하여 조인 연산을 이용하여 질의 결과를 반환하였지만, 이미 I_CLA 관계로 정의 되어 있기 때문에 질의 결과를 빠르게 반환할 수 있다.

유형 3.

RQL	{:\$C} = {:\$D} U {:\$E},
Relations	C_EXT(c, x), C_EXT(d, y), C_EXT(e, z), U_CLA(c, d, e)

위의 질의 유형은 클래스들간의 합집합을 나타내는 U_CLA 관계에 대한 질의 유형이다. <그림 2>에서 'Romantic' 클래스와 'S.F' 클래스의 합집합 연산을 수행하면 'Genre' 클래스가 된다. 즉, 사용자가 "모든 장르의 영화를 검색하라"라는 질의를 수행할 경우, U_CLA 관계를 이용하여 RQL에서 Union 연산의 수행에 소요되는 시간을 줄일 수 있기 때문에 효율적이다.

4. 결론 및 향후 연구

본 논문에서는 ICS-FORTH 프로젝트에서 온톨로지 통합의 대상이 되었던 관계형 데이터베이스와 XML 뿐만 아니라 최근 온톨로지 기술언어로 제안된 OWL을 통합의 대상으로 하기 위하여 뷰 생성 시 사용되는 릴레이션을 제안하였다. 제안한 릴레이션은 기존 뷰로 표현할 수 있는 관계를 확장한 형태로 온톨로지 구성요소 간 다양한 관계에 대하여 기술이 가능하기 때문에 조금 더 다양한 형태의 질의 처리가 가능하다.

향후 연구로는 추가적인 추론이 가능한 OWL-Full의 속성들에 대한 고려가 필요할 것이다.

<참고문헌>

- [1] Michael K. Smith. et al, "OWL Web Ontology Language Guide", W3C, 2004.
- [2] 오삼균, "Web Ontology Language와 그 활용에 관한 고찰", 데이터베이스연구, 18권 3호, 2002. 9.
- [3] Christophides, V. et al, "The ICS-FORTH SWIM: A Powerful Semantic Web Integration Middleware", In Proceedings of the First International Workshop on Semantic Web and Databases (SWDB), Co-located with VLDB 2003.
- [4] G. Karvounarakis. et al, "Querying the Semantic Web with RQL", Computer Networks. 42(5):617-640. August, 2003.
- [5] A. Magkanaraki, V. Tannen, V. Christophides, and D. Plexousakis. "Viewing the Semantic Web Through RVL Lenses", ISWC, 2003.