

온톨로지 통합 프로세스

이인근 황도삼^o
영남대학교 전기공학과, 전자정보공학부^o
iklee@ynu.ac.kr dshwang@yu.ac.kr^o

An Ontology Integration Process

In K. Lee, Dosam Hwang^o
Dept. of Electrical Engineering, Yeungnam University
Dept. of Computer Science, Yeungnam University^o

요 약

온톨로지의 재사용을 위한 온톨로지 통합 연구가 진행 중이다. 기존의 온톨로지 통합 방법으로는 세 개 이상의 온톨로지를 한 번에 통합할 수 없으며, 다수의 작업자가 동시에 통합 작업을 할 수 없다는 단점이 있다. 따라서 본 논문에서는 메타 온톨로지를 이용하여 다수의 작업자가 다수의 온톨로지를 동시에 통합할 수 있는 온톨로지 통합 프로세스(OntoIntegProcess¹⁾)를 제안하고, 온톨로지 통합 실험을 통해 제안한 온톨로지 통합 프로세스의 효용성을 확인한다.

1. 서론

온톨로지 재사용의 필요성에 따라 다수의 소스 온톨로지(source ontology)를 하나의 온톨로지로 통합하는 방법에 관한 연구가 있었다[3-10]. 지금까지 연구되고 개발된 온톨로지 통합 방법 및 통합 도구들을 이용하여 소스 온톨로지를 쉽게 통합할 수 있다. 그러나 통합된 온톨로지의 개념과 구조는 소스 온톨로지와 크게 달라지지 않는다. 그러므로 소스 온톨로지와 구축하려는 온톨로지의 설계, 즉 온톨로지의 사용 목적에 따른 온톨로지의 개념 및 구조가 다른 경우에는 소스 온톨로지를 통합하여 원하는 온톨로지를 구축하기가 쉽지 않다. 즉, 구축하려는 온톨로지의 사용 분야와 관점에 따라 동일한 개념이 다양하게 정의될 수 있고, 소스 온톨로지의 구조와 달리 구축하려는 온톨로지의 구조 또한 새롭게 구성될 수 있다. 그러므로 구축하려는 온톨로지의 설계가 소스 온톨로지와 유사한 경우에는 통합이 쉽게 이루어질 수 있으나, 온톨로지 설계가 다른 경우에는 통합이 쉽지 않다. 따라서 소스 온톨로지와 구축하려는

온톨로지의 설계에 따라 기존에 연구되고 개발된 온톨로지 통합 방법을 적절하게 선택하여 사용한다.

온톨로지를 통합하기 위해 Chimaera, Ontomorph, ONION, OntoMerge, PROMPT 등과 같은 많은 온톨로지 통합 도구들이 개발되어 사용되고 있다[3]. 이들 온톨로지 통합 도구들은 온톨로지 통합 작업자의 온톨로지 통합 과정을 지원하기 위한 도구이다. 이들 온톨로지 통합 도구들은 소스 온톨로지를 참조하여 개념 간의 상하관계와 개념의 속성을 기반으로 유사 개념의 후보를 온톨로지 통합 작업자에게 제시한다. 그러나 온톨로지 통합을 위한 기존의 온톨로지 통합 방법 및 통합 도구는 여러 개의 소스 온톨로지를 한 번에 통합하기 어렵고, 다수의 작업자가 동시에 통합 작업을 하기가 어려운 단점이 있다.

본 논문에서는 메타 온톨로지(meta-ontology)를 이용하여 다수의 온톨로지 통합 작업자가 다수의 온톨로지를 동시에 통합할 수 있는 온톨로지 통합 프로세스를 제안하고, 온톨로지 통합 실험을 통해 제안한 프로세스의 효용성을 확인한다.

2. 관련 연구

온톨로지 통합(integration)은 통합 방법에 따라

1) Ontology Integration Process의 약자임

* 본 논문은 정통부 및 정보통신연구진흥원의 정보통신선도 기반기술개발사업의 연구결과로 수행되었습니다.

Merging, Transformation, Alignment, Articulation 으로 구분할 수 있다[3]. Merging은 서로 다른 두 온톨로지를 조합하여 하나의 새로운 온톨로지를 만드는 것이고, Transformation은 기존에 구축되어 사용되고 있는 온톨로지를 확장하는 것이다. Alignment는 온톨로지 매핑(mapping)이라고도 하는데, 기존 온톨로지의 형태는 유지되고 온톨로지 간의 연결 정보가 추가되는 것이다. 그리고 Articulation은 온톨로지 매핑을 통해 새로운 온톨로지를 생성하는 것이다. 그림 1은 온톨로지 통합의 종류를 그림으로 표현한 것이다.

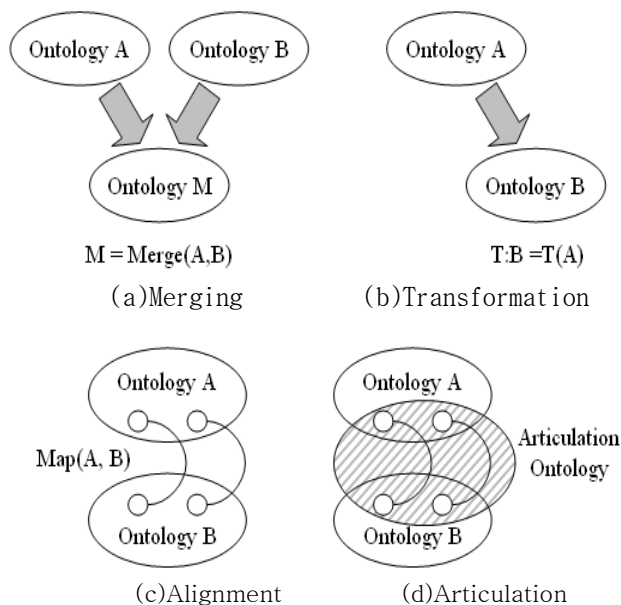


그림 1. 온톨로지 통합 방법의 개념
Fig. 1. Concepts of ontology integration

온톨로지 통합에 관한 연구로써 Hitzler[4]는 온톨로지 통합 과정에서 개념 간의 관계를 통합함에 있어 카테고리 이론을 이용한 온톨로지 통합 개념을 제안하였고, Fridman[5]은 OKDB를 확장한 통합 연산자를 이용하고, synonym, shared substring, common prefixes, common suffixes를 이용하여 개념명 간의 유사성을 계산하였다. 그리고 Deen[6]은 시소러스에 기반하여 동의어를 통합하는 방법을 이용하였다. FCA-Merge[8]는 Formal Concept Analysis라는 수학 기법을 이용하여 소스 온톨로지 개념 간의 관계(동의관계, 상하관계)를 격자로 나타내고 개념 간의 연결 가능성 제시한다. 또한 개념 정의문으로부터 자연언어처리를 통하여 속성 정보를 추출하여 관계성을 결정하는 방법을 제시한다.

온톨로지 통합에 관하여 Chimaera, Ontomorph, ONION, OntoMerge, PROMPT 등과 같은 많은 도구들이 개발되었다. Chimaera[7]는 Ontolingua ontology editor에 기반한 대화식 통합 도구로서, 서로 다른 형태의 소스 온톨로지 통합이 가능하다. 그리고 소스 온톨로지에서의 통합을 위한 개념의 상하 관계를 제시하며, 온톨로지의 개념 통합 시 소스 온톨로지의 정보를 포함하여

보여준다. Ontomorph[9]는 온톨로지 통합 작업자가 소스 온톨로지 사이에서의 개념 쌍을 참조하여 소스 온톨로지의 차이점을 분석하기 위한 연산자를 정의하고, 정의한 연산자를 통해 통합 작업을 수행한다. ONION[10]은 온톨로지 간의 articulation rule을 정의하기 위한 도구를 제공한다. 또한 개념명 간의 유사성을 찾기 위해 사전 정보를 이용하고, 개념명과 코퍼스 단어와의 공기관계(co-occurrence)를 이용한다. PROMPT[3]는 Protégé[11]에서 plug-in 형태로 사용할 수 있는 온톨로지 통합 도구로써, 두 소스 온톨로지의 개념명 사이의 유사도를 계산하여 유사한 개념쌍을 제시한다. 그리고 두 소스 온톨로지에서의 개념(class), 개체(instance), 속성(slot)에 대해 융합(merging), 복사(copy)의 연산을 통해 온톨로지를 통합한다.

3. 제안하는 온톨로지 통합 프로세스

3.1. 온톨로지 구성 요소 정의

온톨로지는 개념(concept, class), 속성(property, slot, role), 제약사항(restriction, facet), 개체(instance, individual)로 구성되어 있는데[2], 온톨로지마다 그 명칭이 다르다²⁾. 따라서 다양한 온톨로지 표현 언어³⁾로 구성된 소스 온톨로지를 통합하기 위해서는 각기 다른 언어로 표현된 소스 온톨로지를 동일한 형태로 통일할 필요가 있다. 즉, 온톨로지 통합을 위해 통일된 형태의 온톨로지 요소를 정의하고, 정의한 온톨로지 요소를 "메타 온톨로지 요소(meta-ontology element)"라고 한다. 그리고 온톨로지 통합 전처리 과정으로 소스 온톨로지 요소를 메타 온톨로지 요소로 변환하여 메타 온톨로지를 구축한다.

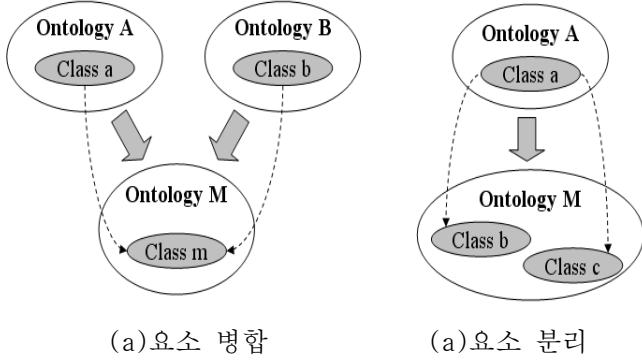
메타 온톨로지 요소는 meta-class, meta-property, meta-relation으로 정의한다. 소스 온톨로지에서의 개념과 개체는 meta-class로 변환되며, 속성은 meta-property로 변환된다. 그리고 개념, 개체, 속성, 제약사항의 조합으로 meta-relation을 구성한다.

그림 2(a)와 같이 서로 다른 소스 온톨로지에서도 동일한 개념이 존재할 경우 각 소스 온톨로지에서의 동일한 개념을 모아 하나의 개념으로 병합해야 한다. 즉, 여러 개의 메타 온톨로지 요소를 그 의미에 기반하여 하나의 메타 온톨로지 요소로 통합하는 것을 "메타 온톨로지 요소 병합(meta-ontology element merging)"이라 한다. 그림 2(b)와 같이 하나의 개념이 나타내는 의미가 목적에 따라 서로 다른 개념으로 나뉠 수 있는 경우에는 하

2) 온톨로지를 구성하기 위한 요소들을 온톨로지 구성 요소(ontology elements)라 하고, 온톨로지 구성 요소에서 개념은 class, concept, 속성은 property, slot, role, 제약사항은 restriction, facet, 개체는 instance, individual로 나타낸다. 온톨로지의 표현 방법이나 종류에 따라 다른 명칭을 사용하기도 하지만 그 의미는 같다.

3) OWL(Ontology Web Language)이나 RDF(Resource Description Framework) 등과 같이 다양한 형태의 온톨로지 표현 언어를 이용하여 온톨로지가 구성된다.

나의 개념을 둘 이상의 개념으로 분리해야 한다. 즉, 하나의 메타 온톨로지 요소를 그 의미에 기반 하여 여러 개의 메타 온톨로지 요소로 분리하는 것을 "메타 온톨로지 요소 분리(meta-ontology element separation)"이라 한다. 그리고 메타 온톨로지 요소의 병합과 분리를 "메타 온톨로지 요소 통합(meta-ontology element integration)"이라고 정의한다.



(a)요소 병합 (a)요소 분리
그림 2. 메타온톨로지 요소 통합

메타 온톨로지 요소는 소스 온톨로지 요소 각각의 공통된 정보를 나타낸다. 메타 온톨로지 요소에는 통합 과정에서 참조한 소스 온톨로지 요소의 정보를 함께 나타내어 통합된 온톨로지 요소의 근원을 알 수 있도록 한다. 표 1에 메타 온톨로지의 형식과 예를 보인다. 표 1에서 Source Element는 메타 온톨로지 요소를 정의할 때, 온톨로지 요소 통합에 참조한 소스 온톨로지 요소의 정보를 기록한 것이다.

표 1(a)은 meta-class form을 나타내는데, 서로 다른 소스 온톨로지(O_1, O_2)에서 같은 개념을 나타내는 'digital_TV'와 'DTV'를 병합하여 meta-class인 'digital_television'을 정의 하고, 소스 온톨로지로부터 한글 대역어, 정의문을 참조하여 'digital_television'의 정보를 표현하였다. 그리고 병합을 위해 참조한 소스 온톨로지의 요소인 'digital_TV(O_1)'와 'DTV(O_2)'를 'digital_television'의 정보에 추가한다.

표 1(b)는 meta-property form을 나타내는데, 소스 온톨로지(O_1, O_2)에서 같은 의미의 속성을 나타내는 'is-a'와 'subclassOf'를 병합하여 meta-property인 'isSubclassOf'를 정의하였다. 그리고 병합을 위해 참조한 소스 온톨로지의 요소인 'is-a(O_1)'와 'subclassOf(O_2)'를 'isSubclassOf'의 정보에 추가한다.

표 1(c)는 meta-relation form을 나타내는데, 소스 온톨로지(O_1, O_2)에서의 restriction, ontology triple을 메타 온톨로지의 meta-class와 meta-property를 참조하여 같은 관계를 나타내는 'digital_TV \exists is-a television(O_1)'와 'DTV \exists subclassOf television(O_2)'를 병합하여 meta-relation인 'digital_television \exists isSubclassOf television'을 설정한다.

표 1. 메타 온톨로지의 형식과 예
(a) meta-class form

Class	digital_television
Korean	디지털 TV
Definition	흑백시대 · 컬러시대를 거친, 이른바 제3세대 텔레비전이다.
Source Element	digital_TV(O_1), DTV(O_2)

(b) meta-property form

Property	isSubclassOf
Definition	개념들의 상하 관계
Source Element	is-a(O_1), subclassOf(O_2)

(c) meta-relation form

Domain	digital_television
Restriction	someValuesFrom (\exists)
Property	isSubclassOf
Range	television
Source Element	digital_TV \exists is-a television(O_1) DTV \exists subclassOf television(O_2)

3.2. 온톨로지 통합 프로세스

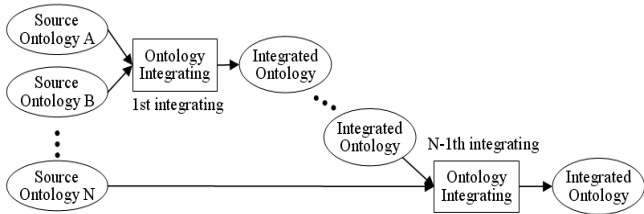
기존의 온톨로지 통합 방법은 두 개의 소스 온톨로지를 통합하여 하나로 만든다. 따라서 여러 개의 소스 온톨로지를 통합하기 위해서는 여러 번의 온톨로지 통합 작업이 필요하다. 즉, N 개의 소스 온톨로지를 통합하기 위해서는 순차적으로 $N-1$ 번의 온톨로지 통합 작업이 필요하다. 그리고 두 개의 소스 온톨로지를 통합하는 평균 시간이 T 일 경우, N 개의 온톨로지를 통합하는데 소요되는 시간은 $(N-1)T$ 가 된다. 그림 3(b)와 같이 병렬로 온톨로지 통합 작업을 하더라도 온톨로지 통합에 소요되는 시간은 최소 $(\log_2 N)T$ 이다. 그러므로 다수의 소스 온톨로지를 통합하기 위해서는 반복적인 온톨로지 통합 과정을 거쳐야 한다.

본 논문에서는 기존의 온톨로지 통합 방법의 단점을 개선하기 위해 다음과 같이 네 단계의 온톨로지 통합 프로세스를 제안한다.

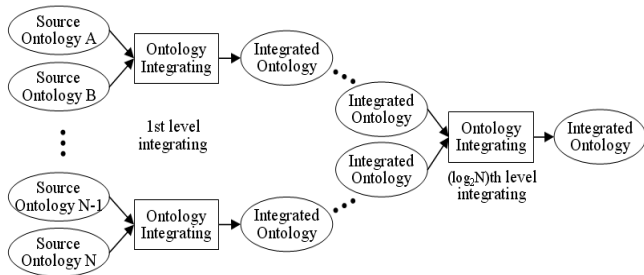
▶ OntoIntegProcess

- [step1] 다양한 온톨로지 표현 언어로 구성된 소스 온톨로지를 메타 온톨로지로 변환한다.
- [step2] 메타 온톨로지를 모두 취합하여 하나의 취합된 메타 온톨로지(mixed meta-ontology)를 구성한다.
- [step3] 취합된 메타 온톨로지로부터 온톨로지 요소의 통합 과정을 거쳐 통합된 메타 온톨로지(integrated meta-ontology)를 구성한다.
- [step4] 통합된 메타 온톨로지를 OWL 형태의 온톨로지로 자동 변환한다.

제안한 온톨로지 통합 프로세스는 한 번의 온톨로지 통합 작업으로 다수의 소스 온톨로지를 통합할 수 있다. 또한, 다수의 온톨로지 통합 작업자가 동시 작업이 가능하다. 그러므로 N 개의 소스 온톨로지 통합하는 시간이 T' 일 경우, M 명의 온톨로지 통합 작업자가 통합 작업을 한다면 소요되는 시간은 최소 T'/M 이다.



(a) 온톨로지 순차 통합 작업



(b) 온톨로지 병렬 통합 작업

그림 3. 기존의 온톨로지 통합 과정

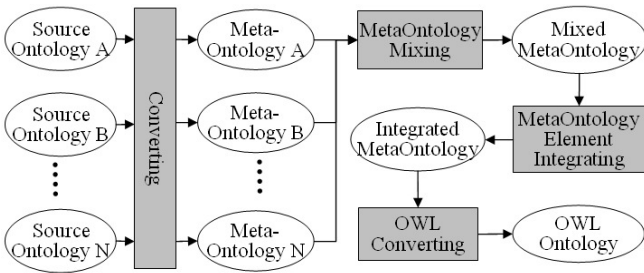


그림 4. 제안한 온톨로지 통합 프로세스

3.2.1. 소스 온톨로지를 메타 온톨로지로 변환

각각의 소스 온톨로지를 메타 온톨로지로 변환한다. 온톨로지 변환은 온톨로지 요소의 변환을 말하며, 온톨로지 요소의 변환은 표 2와 같다. 그리고 그림 5는 온톨로지 요소 변환의 개념을 보인다.

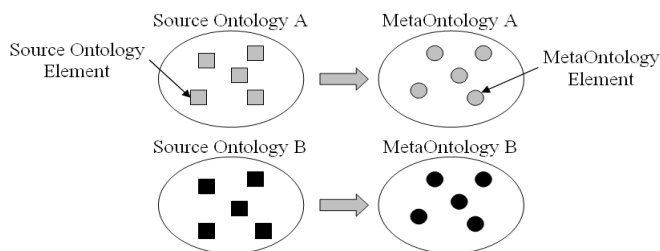


그림 5. 온톨로지 요소 변환

표 2. 온톨로지 요소 변환

소스 온톨로지 요소	메타 온톨로지 요소
▶ class, concept	meta-class
▶ instance, individual	
▶ property, slot, role	meta-property
▶ facet, restriction	meta-relation
▶ ontology triple	
▶ concept-property-concept or property(concept, concept) ▶ instance-property-instance or property(instance, instance)	

3.2.2. 메타 온톨로지 취합

각각의 소스 온톨로지로부터 변환된 메타 온톨로지들 하나의 메타 온톨로지 취합(mixing)한다. 취합은 meta-class 취합, meta-property 취합, meta-relation 취합으로 구분한다. 메타 온톨로지의 취합 과정에서 meta-class와 meta-property는 같은 이름을 가진 메타 온톨로지 요소가 존재할 수 있다. 따라서 같은 이름의 메타 온톨로지 요소를 구분하기 위해 각 요소마다 고유 코드를 부여하여 구분한다. 그리고 meta-relation은 meta-class, meta-property, restriction 등의 조합으로 구성되어 있으므로 domain, range, property와 같은 meta-relation의 정보는 메타 온톨로지의 meta-class와 meta-property의 고유 코드로써 나타낸다.

그림 6은 메타 온톨로지 취합 과정의 개념을 보인다.

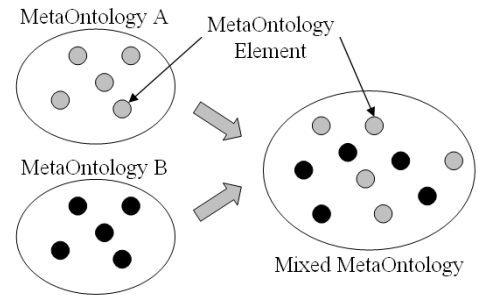


그림 6. 메타 온톨로지 취합

3.2.3. 메타 온톨로지 요소 통합

메타 온톨로지 요소 통합 방법은 1) 동일 형태의 온톨로지 요소 병합, 2) 다른 형태의 온톨로지 요소 병합, 3) 다중 개념의 온톨로지 요소 분리와 같이 크게 세 가지로 구분한다.

1) "동일 형태의 온톨로지 요소 통합"은 서로 다른 소스 온톨로지에서도 같은 의미의 온톨로지 요소가 동일하게 표기된 경우 이를 하나의 메타 온톨로지 요소로 병합하는 것을 말한다. 예를 들어 온톨로지 (O_1, O_2) 에서 정의된 두 개의 개념 'digital_television(O_1)'과 'digital_television(O_2)'이 동일한 개념이고, 'digi-

tal_television'과 같이 개념명이 동일하게 표기되었을 경우 이를 병합하여 'digital_television(O_M)'으로 정의하는 경우이다.

2)"다른 형태의 온톨로지 요소 병합"은 서로 다른 소스 온톨로지에서 동일한 의미의 온톨로지 요소의 표기가 다른 경우, 이를 하나의 메타 온톨로지 요소로 병합하는 것을 말한다. 예를 들어 온톨로지(O_1, O_2)에서 정의된 두 개의 개념 'digital_TV(O_1)'와 'DTV(O_2)'가 동일한 개념이지만 서로 다르게 표기된 경우, 이를 통합하여 'digital_television(O_M)'으로 정의하는 경우이다.

3)"다중 개념의 온톨로지 요소 분리"는 소스 온톨로지에서 정의된 온톨로지 요소가 그 목적에 따라 두 개의 개념으로 정의되어야 하는 경우, 이를 두 개의 메타 온톨로지 요소로 분리하는 것을 말한다. 예를 들어 소스 온톨로지(O_1)에서는 'MPEG(O_1)'과 같이 하나의 개념을 정의하였으나 온톨로지 설계에 따라 'MPEG'의 의미를 '집단(group)'과 '방법(method)'으로 나누어야 할 경우, 'MPEG'를 'MPEG_method(O_M)'와 'MPEG_group(O_M)'으로 분리하여 정의하는 경우이다. 또한, 소스 온톨로지(O_1, O_2)에서 서로 다른 의미의 온톨로지 요소가 동일하게 표기되었을 경우 온톨로지 요소를 의미별로 다르게 표기하는 경우도 iii)과 같은 방법으로 메타 온톨로지 요소를 통합한다. 예를 들어 소스 온톨로지(O_1, O_2)에서 정의된 '집단(group)'의 의미의 'MPEG(O_1)'와 '방법(method)'의 의미의 'MPEG(O_2)'가 'MPEG'와 같이 동일하게 표기되었더라도 'MPEG_method(O_M)'와 'MPEG_group(O_M)'와 같이 분리하여 정의한다.

그림 7은 메타 온톨로지 요소 병합의 개념을 보인다.

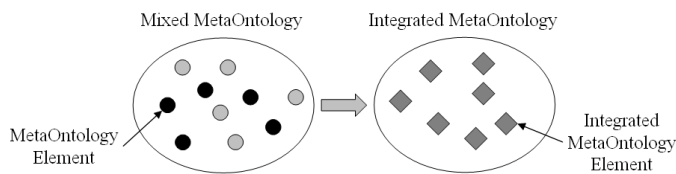


그림 7. 메타 온톨로지 요소 통합

3.2.4. 메타 온톨로지의 OWL 변환

통합된 메타 온톨로지를 OWL[12]로 변환한다. 그림 8은 메타 온톨로지의 OWL 변환 개념을 보인다.

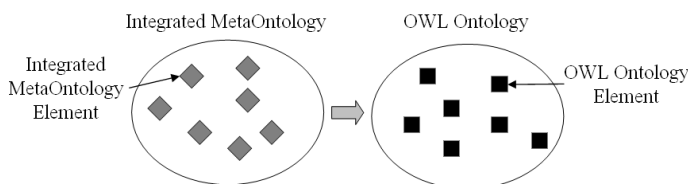
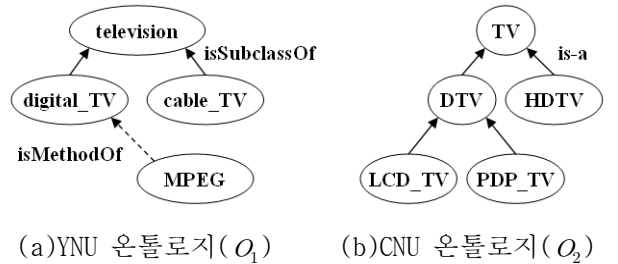


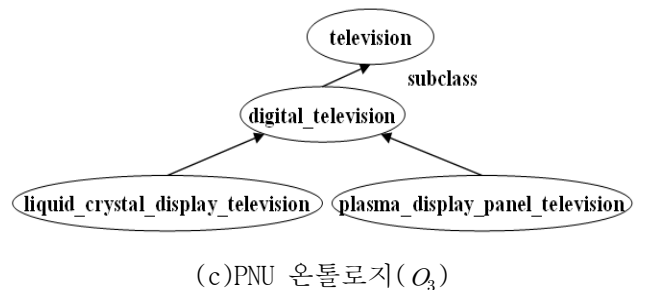
그림 8. 메타 온톨로지의 OWL 변환

4. 온톨로지 통합 실험

온톨로지 통합 실험을 위해 3개의 온톨로지 구축 그룹에서 개별적으로 구축한 온톨로지(O_1, O_2, O_3)를 사용하였다. 그림 9는 각 기관에서 구축한 온톨로지의 일부를 보이는데, 이들은 개념명과 속성명의 표기 방법이 다르나 개념 사이의 상-하 관계 구성은 유사하다.



(a)YNU 온톨로지(O_1) (b)CNU 온톨로지(O_2)

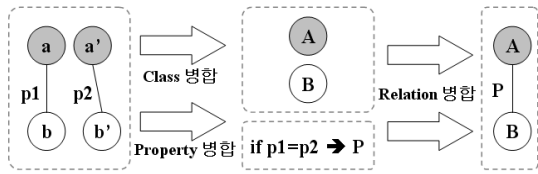


(c)PNU 온톨로지(O_3)

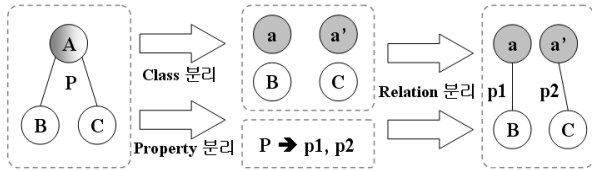
그림 9. 구축된 온톨로지의 일부

본 논문에서는 소스 온톨로지 요소의 일부에 대해 OntoIntegProcess를 통합 과정을 설명한다.

온톨로지 통합 과정의 효율을 높이기 위해, 메타 온톨로지의 통합은 'Class 통합', 'Property 통합', 'Relation 통합'의 순서대로 수행하였다. 그림 10은 메타 온톨로지 요소의 병합과 분리 과정을 보인다. 그림 10(a)는 온톨로지 요소의 병합의 순서를 보인다. 즉, 개념 a 와 개념 a' , 그리고 개념 b 와 개념 b' 가 의미적으로 동일하지만 표기가 다른 경우, 먼저 'Class 병합'을 수행하여 개념 A 와 개념 B 를 정의한다. 그리고 속성 $p1$ 과 속성 $p2$ 가 의미적으로 동일하나 표기가 다른 경우, 'Property 병합'을 수행하여 속성 P 를 정의한다. 또한, $p1(a,b)$ 와 $p2(a',b')$ 관계를 'Relation 병합'을 수행하여 정의된 개념과 속성으로 $P(A,B)$ 관계를 설정한다. 그림 10(b)는 온톨로지 요소의 분리 과정을 보인다. 즉, 개념 A 의 의미가 온톨로지 설계에 따라 두 개의 개념으로 분리 가능한 경우, 먼저 'Class 분리'를 수행하여 개념 a 와 a' 를 정의한다. 그리고 속성 P 또한 'Property 분리'를 수행하여 $p1$ 과 $p2$ 를 정의한다. $P(A,B)$ 와 $P(A,C)$ 관계를 'Relation 분리'를 수행하여 새롭게 정의한 개념과 속성으로 $p1(a,B)$ 와 $p2(a',C)$ 관계를 설정한다.



(a) 온톨로지 요소 병합



(a) 온톨로지 요소 분리

그림 10. 온톨로지 요소 통합 예

그림 11은 표 3의 온톨로지 통합 과정을 통해 온톨로지를 통합한 결과를 보인다.

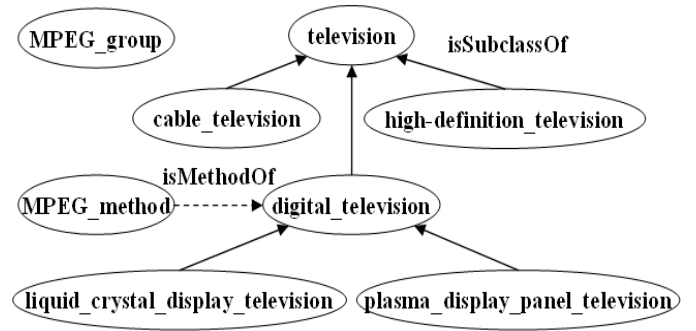


그림 11. 통합된 온톨로지 (O_M)

표 3. 온톨로지 통합 절차

(a) Class 통합

Source Ontology	Operator	Integrated Ontology
television(O_1), TV(O_2), television(O_3)	Merge	television(O_M)
digital_TV(O_1), DTV(O_2), digital_tv(O_3)	Merge	digital_tv(O_M)
cable_TV(O_1)	Merge	cable_tv(O_M)
HDTV(O_2)	Merge	high-definition_tv(O_M)
LCD_TV(O_2), liquid_crystal_display_tv(O_3)	Merge	liquid_crystal_display_tv(O_M)
PDP_TV(O_2), plasma_display_panel_tv(O_3)	Merge	plasma_display_panel_tv(O_M)
MPEG(O_1)	Separate	MPEG_method(O_M), MPEG_group(O_M)

(b) Property 통합

Source Ontology	Operator	Integrated Ontology
isSubclassOf(O_1), is-a(O_2), subclass(O_3)	Merge	isSubclassOf(O_M)
isMethodOf(O_1)	Merge	isMethodOf(O_M)

(a) Class 통합

Source Ontology	Operator	Integrated Ontology
isSubclassOf(digital_TV, television)(O_1) is-a(DTV, TV)(O_2) subclass(digital_tv, television)(O_3)	Merge	isSubclassOf(digital_tv, television)(O_M)
isSubclassOf(cable_TV, television)(O_1)	Merge	isSubclassOf(cable_tv, television)(O_M)
is-a(HDTV, TV)(O_2)	Merge	is-a(high-definition_tv, television)(O_M)
is-a(LCD_TV, DTV)(O_2) subclass(liquid_crystal_display_tv, digital_tv)(O_3)	Merge	isSubclassOf(liquid_crystal_display_tv, digital_tv)(O_M)
is-a(PDP_TV, DTV)(O_2) subclass(plasma_display_panel_tv, digital_tv)(O_3)	Merge	isSubclassOf(plasma_display_panel_tv, digital_tv)(O_M)
isMethodOf(MPEG, digital_TV)(O_1)	Separate	isMethodOf(MPEG_method, digital_tv)(O_M)

5. 결론 및 향후 연구과제

온톨로지의 구축 분야와 사용 목적에 따른 온톨로지 설계와 구축은 많은 시간과 노력을 필요로 한다. 따라서 기존에 구축되어 사용되고 있는 온톨로지를 재사용하여 새로운 온톨로지를 구축하려는 연구가 있었다. 그러나 기존의 온톨로지 통합 방법과 도구들은 소스 온톨로지의 설계와 구축하려는 온톨로지의 설계가 비슷한 경우에는 그 활용도가 높으나, 온톨로지의 설계가 다른 경우에는 온톨로지 통합 작업은 온톨로지 구축 작업 못지않은 시간과 노력이 소요된다. 또한, 기존의 온톨로지 통합 도구들은 다양한 언어로 표현된 다수의 온톨로지를 통합하기 어렵다. 따라서 다양한 설계와 언어로 표현된 다수의 온톨로지를 통합하기 위한 방법이 필요하다.

본 논문에서는 다양한 언어로 표현된 다수의 온톨로지를 통합하기 위해 메타 온톨로지를 정의하였고, 메타 온톨로지를 이용한 온톨로지 통합 프로세스를 제안하였다. 그리고 온톨로지를 통합 실험을 통해 제안한 프로세스의 효용성을 확인하였다.

통합한 온톨로지를 사용하기 위해서는 개념(concept), 개체(instance), 속성(property), 관계(relation) 등의 온톨로지 요소를 정의하는 것 이외에도 제약사항(restriction), 공리(axiom) 등과 같은 온톨로지 추론에 필요한 추가적인 정보가 필요하다. 그러나 추론에 필요한 정보를 한 번의 통합 작업으로 통합된 온톨로지에 반영하는 것은 어려운 일이다. 즉, 온톨로지의 구조적인 통합 후 별도의 작업을 통해 개념에 필요한 제약사항을 설정한다. 따라서 소스 온톨로지의 추론 설계에 대한 정보 통합이 가능한 온톨로지 통합 방법에 대한 연구가 필요하다.

참고 문헌

[1] 이인근, 황도삼, 서석태, 권순학, "메타온톨로지에 기반한 온톨로지의 통합", 한국퍼지 및 지능 시스템학회 논문지, 게재예정.

[2] Natalya Fridman Noy, and Deborah L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," Stanford Knowledge Systems Laboratory Technical Report KSL-01-05, March 2001.

[3] Natalya F. Noy, Mark A. Musen, "The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping," International Journal of Human-Computer Studies, Vol. 59, No. 6, pp. 983-1024, December 2003.

[4] Hitzler, Kroetz, Ehrig, Sure, "What is Ontology Merging?," Workshop at the 20th National Conference on Artificial Intelligence, AAAI-05, Pittsburgh, Pennsylvania, July 2005.

[5] N. Fridman Noy, M. A. Musen, "An Algorithm for Merging and Aligning Ontologies: Automation and Tool Support," In Proc. of the Workshop on Ontology Management at the Sixteenth National Conference on Artificial Intelligence (AAAI-99), Orlando, 1999.

[6] S. M. Deen, K. Ponnampuruma, "Dynamic Ontology Integration in a Multi-agent Environment," Proc. of the 20th International Conference on Advanced Information Networking and Applications (AINAI'06), pp. 373-378, 2006.

[7] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder, "An Environment for Merging and Testing Large Ontologies," Proc. 7th Int'l Conf. Principles of Knowledge Representation and Reasoning (KR 00), Morgan Kaufmann, pp. 483-493, 2000.

[8] G. Stumme and A. Maedche, "FCA-MERGE: Bottom-up merging of ontologies," In IJCAI, pp. 225-234, 2001.

[9] H. Chalupsky, "OntoMorph: A Translation System for Symbolic Knowledge," Proc. 7th Int'l Conf. Principles of Knowledge Representation and Reasoning (KR 2000), Morgan Kaufmann, 2000.

[10] P. Mitra, G. Wiederhold and S. Decker, "A Scalable Framework for Interoperation of Information Sources," The first Semantic Web Working Symposium, pp. 317-330, 2001.

[11] Protégé, <http://protege.stanford.edu>, January 2007.

[12] OWL Web Ontology Language Guide, <http://www.w3.org/TR/owl-guide>, January 2007.