

메타온톨로지에 기반한 온톨로지의 통합

Ontology Integration based on MetaOntology

이인근, 황도삼*, 서석태, 권순학*

In K. Lee, Dosam Hwang, Suk T. Seo, Soon H. Kwon

영남대학교 전기공학과

** 영남대학교 전자정보공학부

요 약

온톨로지의 재사용을 위해 온톨로지를 통합하는 방법에 대한 연구가 진행되고 있다. 그리고 다양한 온톨로지 통합 도구들이 개발되어 사용되고 있다. 그러나 기존의 온톨로지 통합 방법 및 도구는 셋 이상의 온톨로지를 한번에 통합할 수 없으며, 다수의 작업자가 동시에 통합 작업을 할 수 없는 단점이 있다. 본 논문에서는 메타온톨로지를 이용한 온톨로지 통합 프로세스(OntoIntegProcess¹⁾)를 제안하고, 온톨로지 통합 프로세스에 기반하여 온톨로지 통합 시스템(OntoIS²⁾)을 개발한다. 온톨로지의 통합 실험을 통해 제안한 프로세스와 시스템의 효용성을 확인한다.

키워드 : 온톨로지, 통합, 프로세스, 메타온톨로지

Abstract

Researches are working on integration of ontology in order to reuse the constructed ontologies. And lots of tools for ontology integration have been developed and used. However conventional methods and tools have following shortages: they cannot integrate more than three ontologies at once, and numbers of people cannot work together for ontology integration with the tools. In this paper, we propose an ontology integration process (OntoIntegProcess¹⁾) by using MetaOntology[1]. And we develop an ontology integration system (OntoIS²⁾) based on the process. We show the effectiveness of the proposed process and system through experimental results obtained from ontology integration.

Key Words : ontology, integration, process, MetaOntology

1. 서 론

온톨로지는 구축 분야와 사용 목적에 따라 설계되며, 설계된 온톨로지에 적합한 구축 방법으로 온톨로지가 구축된다 [1]. 그러나 온톨로지는 설계와 구축에 많은 시간과 노력을 필요로한다. 이런 시간과 노력을 줄이기 위해서는 기존에 구축되어 사용되고 있는 온톨로지, 즉, 소스 온톨로지(source ontology)를 재사용할 필요가 있다. 온톨로지 재사용의 필요성에 따라 다수의 소스 온톨로지를 하나의 온톨로지로서 통합하는 방법에 관한 연구가 진행되고 있다.

온톨로지 통합을 위해 Chimaera, Ontomorph, ONION, OntoMerge, PROMPT 등과 같은 온톨로지 통합 도구들이 개발되어 사용되고 있다[3]. 이들 도구들은 개념간의 상·하관계와 개념의 속성을 기반으로 유사 개념의 후보를 제시함으로써 온톨로지의 통합 과정을 지원한다. 그러나 기존의 온톨로지 통합 방법과 도구는 소스 온톨로지의 개념과 개념의 계층 구조를 복제하며 통합하기 때문에 통합된 온톨로지는

소스 온톨로지의 설계³⁾를 크게 벗어나지 못한다. 그러므로 소스 온톨로지와 구축하려는 온톨로지의 설계가 다른 경우에는 온톨로지 통합을 통해 원하는 온톨로지를 구축하기가 어렵다.

기존의 온톨로지 통합 방법 및 도구는 i)소스 온톨로지와 구축하려는 온톨로지를 효율적으로 관리하고, ii)온톨로지 통합 과정에서 소스 온톨로지의 개념을 비교하고 검색함으로써 iii)온톨로지 통합 작업자가 온톨로지를 쉽게 통합할 수 있는 환경을 제공한다. 그러나 이들을 이용한다 하더라도, 여러 개의 소스 온톨로지를 한 번에 통합하기 어렵고, 또한 다수의 작업자가 동시에 통합 작업을 하기가 어려운 단점이 있다.

본 논문에서는 다수의 작업자가 동시에 다양한 언어로 구성된 온톨로지를 동시에 통합하기 위한 방법을 제안한다. 제안한 온톨로지 통합 방법의 구현을 위해 메타온톨로지[1]를 이용한 온톨로지 통합 프로세스를 구성하고, 제안한 온톨로지 통합 프로세스에 기반하여 온톨로지 통합 환경을 구축한다.

접수일자 : 2007년 6월 2일

완료일자 : 2007년 7월 21일

*본 논문은 정통부 및 정보통신연구진흥원의 정보통신 선도기반기술개발사업의 일부 지원으로 수행되었습니다.

*Corresponding author

1) Ontology Integration Process의 약자임

2) Ontology Integration System의 약자임

3) 온톨로지를 구축하기 전 온톨로지의 구축 목적에 따라 온톨로지를 설계한다. 온톨로지 설계에 따라 개념 정의, 개념의 계층 구조, 개념 간의 관계 설정 기준이 달라진다.

2. 관련 연구

온톨로지 통합(integration)은 방법에 따라 'Merging', 'Transformation', 'Alignment', 'Articulation' 으로 구분할 수 있다[3]. 'Merging'은 서로 다른 두 온톨로지를 조합하여 하나의 새로운 온톨로지를 만드는 것이고, 'Transformation'은 기존에 구축되어 사용되고 있는 온톨로지를 확장하는 것이다. 'Alignment'는 '온톨로지 매핑(ontology mapping)'이라고도 하는데, 기존 온톨로지의 형태는 유지되고 온톨로지 간의 연결 정보가 추가되는 것이다. 그리고 'Articulation'은 '온톨로지 매핑'을 통해 새로운 온톨로지를 생성하는 것이다. 그림 1은 이들 온톨로지 통합 방법의 개념을 보인다.

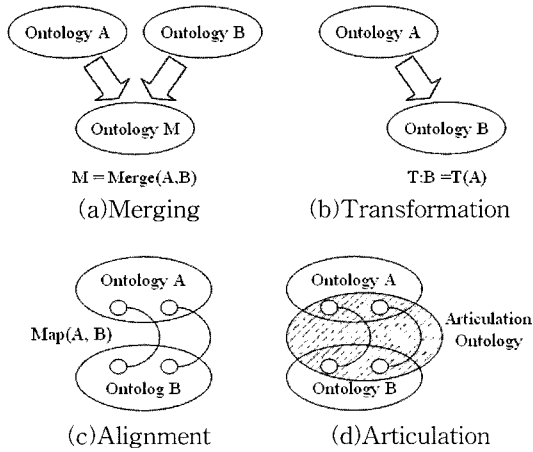


그림 1. 온톨로지 통합 방법의 개념
Fig. 1. Concepts of ontology integration

온톨로지 통합을 위해, Hitzler[4]는 개념 간의 관계를 통합함에 있어 카테고리 이론을 이용한 온톨로지 통합의 개념을 설명하였고, Fridman[5]은 OKDB를 확장한 통합 연산자를 사용하였고, 'synonym', 'shared substring', 'common prefixes', 'common suffixes'를 이용한 개념명 간의 유사성을 계산하였다. 그리고 Deen[6]은 시소러스에 기반하여 동의어를 통합하는 방법을 이용하였다. FCA-Merge[8]는 'Formal Concept Analysis'라는 수학기법을 이용하여 온톨로지의 개념 관계(동의관계, 상·하관계)를 격자로 나타내고 개념 연결 가능성 제시하였다. 또한 개념 정의문으로부터 자연언어처리를 통하여 속성정보를 추출하고 관계성을 결정하는 방법을 제시하였다.

그리고, Chimaera, OntoMorph, ONION, OntoMerge, PROMPT 등과 같은 온톨로지 통합 도구들이 개발되었다. Chimaera[7]는 'Ontolingua ontology editor'에 기반한 대화식 통합 도구로서, 서로 다른 형태의 소스 온톨로지 통합이 가능하다. 그리고 소스 온톨로지에서의 통합을 위한 개념의 상·하 관계를 제시하며, 온톨로지의 개념 통합시 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 rule'을 정의하기 위한 도구를 제공한다. 또한, 개념명간의 유사성을 찾기 위해 사전 정보를 이용하고, 개념명과 코퍼스 단어와의 '공기관계(co-occurrence)'를 이용한다. PROMPT[3]는 Protégé[12]에

서 plug-in 형태로 사용할 수 있는 온톨로지 통합 도구로서, 두 소스 온톨로지의 개념명 사이의 유사도를 계산하여 유사한 개념 쌍을 제시한다. 그리고 두 소스 온톨로지에서의 'class', 'instance', 'slot'에 대한 '병합(merging)', '복사(copy)' 등의 연산을 통해 온톨로지를 통합한다.

3. 온톨로지 통합

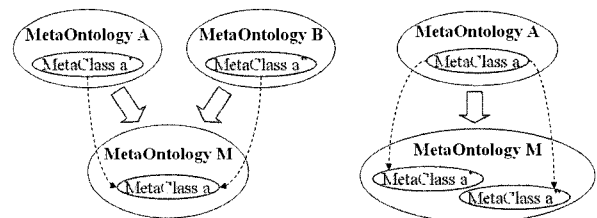
본 논문에서는 그림 1(a)와 같이 서로 다른 형태의 소스 온톨로지를 조합하여 하나의 온톨로지로 변환하는 온톨로지 'Merging'을 목적으로 하며, 'Merging' 과정에서 'Transformation', 'Alignment', 'Articulation'이 가능한 온톨로지 통합 방법을 제안한다.

3.1 온톨로지 구성 요소 정의

온톨로지는 개념(concept, class), 속성(property, slot, role), 개체(individual, instance), 제약사항(restriction, facet)으로 구성되어 있는데, 온톨로지의 종류에 따라 그 명칭이 다르다⁴⁾. 따라서 다양한 온톨로지 언어⁵⁾로 표현된 소스 온톨로지의 통합 진단개로 "메타온톨로지 요소(MetaOntology element)"라는 통일된 형태의 온톨로지 구성 요소를 정의한다.

'메타온톨로지 요소'는 MetaClass, MetaProperty, MetaRelation으로 구분한다. 소스 온톨로지에서의 개념과 개체는 MetaClass로 변환되며, 속성은 MetaProperty로 변환된다. 그리고 개념, 개체, 속성, 제약사항의 조합으로 MetaRelation을 구성한다.

그림 2(a)와 같이 서로 다른 소스 온톨로지에서의 동일한 개념이 존재할 경우 동일한 개념들을 하나의 개념으로 병합한다. 그리고 여러 개의 '메타온톨로지 요소'를 하나로 통합하는 것을 "메타온톨로지 요소 병합(Merging of MetaOntology elements)"이라고 정의한다. 그림 2(b)와 같이 한 개념의 의미가 서로 다른 개념으로 구분하여 표현하고자 할 경우, 하나의 '메타온톨로지 요소'를 여러 개의 '메타온톨로지 요소'로 분리한다. 그리고 이것을 "메타온톨로지 요소 분리(Separation of MetaOntology elements)"라고 정의한다. 그리고 '메타온톨로지 요소'의 병합과 분리를 "메타온톨로지 요소 통합(Integration of MetaOntology elements)"으로 정의한다.



(a)메타온톨로지 요소 병합 (b)메타온톨로지 요소 분리

그림 2. 메타온톨로지 요소 통합

Fig. 2. Integration of MetaOntology elements

- 4) 온톨로지를 구성하기 위한 요소들을 온톨로지 구성 요소(ontology elements)라 하고, 온톨로지 언어에 따라 다른 명칭을 사용하기도 하지만 그 의미는 유사하다.
- 5) OWL이나 RDF(S)등과 같이 다양한 형태의 온톨로지 언어로 온톨로지가 표현된다.

‘메타온톨로지 요소’에는 통합 과정에서 참조한 소스 온톨로지 요소의 정보를 함께 나타내어 통합된 온톨로지 요소의 출처를 알 수 있도록 한다. 표 1에 메타온톨로지의 형식과 예를 보인다. 표 1에서 ‘Source Element’는 온톨로지 요소의 통합 시에 참조한 소스 온톨로지 요소의 정보를 기록한 것이다.

표 1. 메타온톨로지의 형식과 예

Table 1. MetaOntology form and its examples

(a) MetaClass form	
Class	digital_television
Korean	디지털 TV
Definition	흑백시대 · 컬러시대를 거친, 이른바 제3세대 텔레비전이다.
Source Element	digital_TV(O_1), DTV(O_2)
(b) MetaProperty form	
Property	isSubclassOf
Definition	개념의 상 · 하 관계
Source Element	is-a(O_1), subclassOf(O_2)
(c) MetaRelation 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)

표 1(a)는 ‘MetaClass form’을 나타낸다. 예를 들어, 소스 온톨로지(O_1, O_2)에서 개념이 같은 “digital_TV”와 “DTV”를 병합하여 “digital_television”을 정의한다. 그리고 병합을 위해 참조한 소스 온톨로지의 요소인 “digital_TV(O_1)”와 “DTV(O_2)”를 “digital_television”의 ‘Source Element’로 기록한다.

표 1(b)는 ‘MetaProperty form’을 나타낸다. 예를 들어, 소스 온톨로지(O_1, O_2)에서 같은 의미의 속성인 “is-a”와 “subclassOf”를 병합하여 “isSubclassOf”를 정의하였다. 그리고 병합을 위해 참조한 소스 온톨로지의 요소인 “is-a(O_1)”와 “subclassOf(O_2)”를 “isSubclassOf”의 ‘Source Element’로 기록한다.

표 1(c)의 ‘MetaRelation form’은 표 2와 같이 소스 온톨로지(O_1, O_2)에서의 ‘restriction’과 ‘binary relation’을 표현하기 위한 형식이다. 메타온톨로지에서 같은 관계를 나타내는 개념 관계인 “digital_TV \exists is-a television(O_1)”와 “DTV \exists subclassOf television(O_2)”를 병합하여 “digital_television \exists isSubclassOf television”을 설정한다.

3.2 온톨로지 통합 프로세스(OntoIntegProcess)

기존의 온톨로지 통합은 두 개의 온톨로지가 통합되는 방식이므로, 여러 개의 소스 온톨로지를 통합하기 위해서는 여러 번의 온톨로지 통합 작업이 필요하다. 즉, 그림 3(a)와 같이 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. 또한, 그림 3(b)와 같이 병렬로 온톨로지 통합 작업을 하더라도 온톨로지 통합에 소요되는 시간은 최소 $T \log_2 N$ 이다. 그러므로 다수의 소스 온톨로지를 통합하기 위해서는 반복적인 온톨로지 통합 과정을 거쳐야 하는 단점이 있다.

본 논문에서는 기존의 온톨로지 통합 방법의 단점을 개선하기 위해 그림 4와 같은 네 단계의 온톨로지 통합 프로세스를 제안한다.

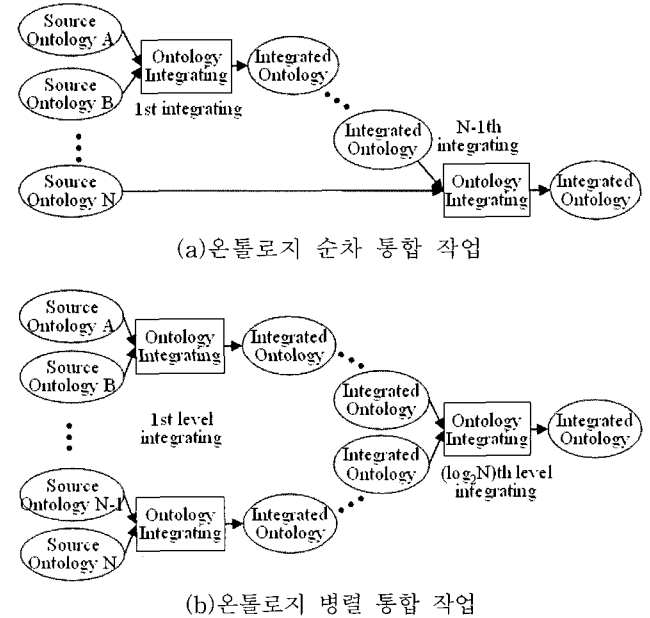


그림 3. 기존의 온톨로지 통합 과정
Fig. 3. Conventional process on ontology integration

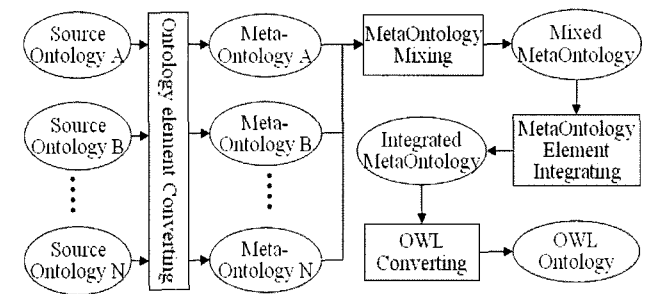


그림 4. 제안한 온톨로지 통합 프로세스
Fig. 4. Proposed process on ontology integration

OntoIntegProcess

- [step1] 다양한 온톨로지 언어로 표현된 소스 온톨로지를 메타온톨로지로 자동 변환한다.
- [step2] 메타온톨로지를 모두 취합하여 하나의 취합된 메타온톨로지(Mixed MetaOntology)를 구성한다.
- [step3] 취합된 메타온톨로지로부터 온톨로지 요소의 통합 과정을 거쳐 통합된 메타온톨로지(Integrated MetaOntology)를 구성한다.
- [step4] 통합된 메타온톨로지를 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.

3.2.1 온톨로지 요소 변환

온톨로지의 변환은 소스 온톨로지 요소를 '메타온톨로지 요소'로 변환하는 것이다. 온톨로지 요소의 변환은 표 2와 같다. 그리고 그림 5는 온톨로지 요소 변환의 개념을 보인다.

표 2. 온톨로지 요소 변환
Table 2. Mapping of ontology elements

소스 온톨로지 요소	메타온톨로지 요소
▶ class, concept ▶ instance, individual	MetaClass
▶ property, slot, role	MetaProperty
▶ facet, restriction ▶ binary relation · concept-property-concept or property(concept, concept) · instance-property-instance or property(instance, instance)	MetaRelation

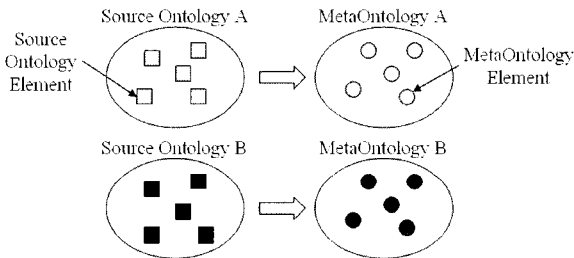


그림 5. 온톨로지 요소의 변환
Fig. 5. Conversion of ontology elements

3.2.2 메타온톨로지 취합

각각의 소스 온톨로지로부터 변환된 메타온톨로지를 하나의 메타온톨로지 취합(mixing)한다. 취합은 MetaClass 취합, MetaProperty 취합, MetaRelation 취합으로 구분한다. 그림 6은 메타온톨로지 취합의 개념을 보인다.

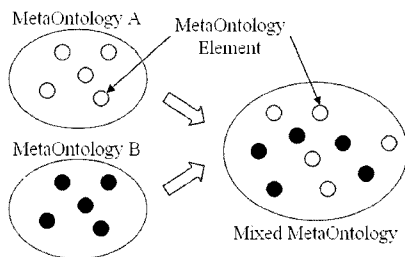


그림 6. 메타온톨로지의 취합
Fig. 6. Mixing of MetaOntology

3.2.3 메타온톨로지 요소의 통합

'메타온톨로지 요소'의 통합 방법은 i)동일 형태의 온톨로지 요소 병합, ii)다른 형태의 온톨로지 요소 병합, iii)다중 개념의 온톨로지 요소 분리와 같이 크게 세 가지로 구분한다.

i) '동일 형태의 온톨로지 요소 병합'은 서로 다른 소스 온톨로지에서 같은 의미의 온톨로지 요소가 동일하게 표기된 경우 이를 하나의 '메타온톨로지 요소'로 병합하는 것을 말한다. 예를 들어, 온톨로지(O_1, O_2)에서 정의된 두 개의 개념 "digital_television(O_1)"과 "digital_television(O_2)"의 의미가 같고, 개념명이 "digital_television"으로 동일하게 표기된 경우 이를 "digital_television(O_M)"으로 병합한다.

ii) '다른 형태의 온톨로지 요소 병합'은 서로 다른 소스 온톨로지서 표기는 다르나 의미가 같은 경우, 이를 하나로 병합하는 것을 말한다. 예를 들어, 온톨로지(O_1, O_2)에서 정의된 두 개의 개념 "digital_TV(O_1)"와 "DTV(O_2)"의 의미가 같은 경우, 이를 "digital_television(O_M)"으로 병합한다.

iii) '다중 개념의 온톨로지 요소 분리'는 하나의 온톨로지 요소가 그 목적에 따라 두 개로 분리하는 것을 말한다. 예를 들어, 소스 온톨로지(O_1)에서 정의된 "MPEG(O_1)"이 목적에 따라 "집단(group)"과 "방법(method)"으로 구분될 필요가 있을 경우, 이를 "MPEG_method(O_M)"와 "MPEG_group(O_M)"으로 분리하여 정의한다. 또한, 소스 온톨로지(O_1, O_2)에서 서로 다른 의미의 온톨로지 요소가 동일하게 표기되었을 경우에는 온톨로지 요소를 의미별로 다르게 표기해야 하는데, 이 경우에도 iii)과 같은 방법을 이용한다. 예를 들어, 소스 온톨로지(O_1, O_2)에서 "집단(group)" 의미의 "MPEG(O_1)"와 "방법(method)" 의미의 "MPEG(O_2)"가 "MPEG"와 같이 동일하게 표기되었더라도 "MPEG_method(O_M)"와 "MPEG_group(O_M)"로 분리하여 정의한다.

그림 7은 '메타온톨로지 요소'의 통합 개념을 보인다.

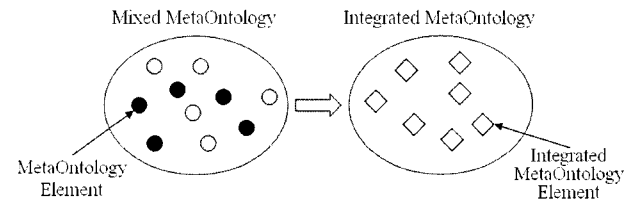


그림 7. 메타온톨로지 요소의 통합
Fig. 7. Integration of MetaOntology elements

3.2.4 메타온톨로지의 OWL 변환

통합된 메타온톨로지를 OWL[13]로 작성된 온톨로지 변환한다. 메타온톨로지의 OWL 변환은 온톨로지 언어만을 OWL로 변환하는 것이다. 그림 8은 메타온톨로지의 OWL 변환 개념을 보인다.

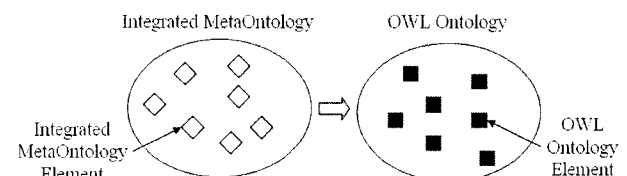


그림 8. 메타온톨로지의 OWL 변환
Fig. 8. OWL conversion of MetaOntology

4. 온톨로지 통합 시스템

온톨로지의 통합 도구 개발의 목적은 소스 온톨로지의 요소를 통합하여 새로운 온톨로지를 쉽게 구축할 수 있도록 하는 것이다. 이를 위해서는 소스 온톨로지의 정보를 손쉽게 검색할 수 있어야 하고, 통합 과정 또한 간단해야 한다. 따라서 본 논문에서는 제안한 온톨로지 통합 프로세스를 기반으로 하며, 온톨로지 통합 과정에서 편리한 기능을 지원하여 온톨로지를 쉽게 통합할 수 있는 온톨로지 통합 시스템을 구축한다. 그리고 온톨로지 통합 연산자를 정의하고, 연산자를 이용하여 온톨로지 통합의 상세 과정을 표현한다.

4.1 온톨로지 통합 시스템(OntoIS)의 구조

OntoIS는 그림 9와 같이 6개의 모듈로 구성되어 있다. i) "MetaOntology 변환기"는 소스 온톨로지를 메타온톨로지 형태로 변환하여 데이터베이스에 저장한다. 소스 온톨로지의 종류와 개수의 제약 없이 MetaOntology로 변환 가능하다. ii) "Mixed MetaOntology 관리 도구"는 취합된 메타온톨로지를 관리할 수 있는 환경을 제공한다. iii) "Integrated MetaOntology 관리 도구"는 통합된 메타온톨로지를 관리하고 그 구조를 확인할 수 있는 도구이다. iv) "MetaOntology 통합 도구"는 취합된 메타온톨로지 통합하여 하나의 통합된 메타온톨로지를 만들기 위한 환경을 제공한다. v) "MetaOntology 통합 추천 도구"는 "MetaOntology 통합 도구"와 연동하여 취합된 메타온톨로지 중에서 통합 가능성이 있는 온톨로지 요소의 후보들을 제시함으로써 온톨로지 통합 효율을 높인다. vi) "OWL 변환기"는 통합된 메타온톨로지를 OWL 형태의 온톨로지 로 자동 변환하여 온톨로지 관련 시스템에서 활용할 수 있도록 한다.

OntoIS는 웹 기반의 온톨로지 통합 시스템으로써 다수의 온톨로지 통합 작업자가 동시에 온톨로지 통합 작업을 가능하게 한다. 그림 10은 OntoIS의 사용자 화면을 보인다.

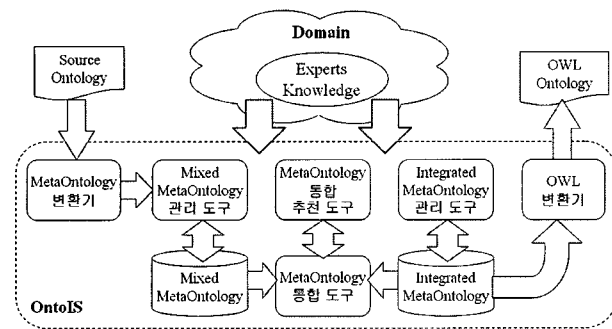


그림 9. OntoIS의 구조
Fig. 9. Configuration of OntoIS

4.2 온톨로지 통합 연산자

본 논문에서는 표 3과 같이 "메타온톨로지 통합 연산자 (Operators for MetaOntology integration)"를 정의한다. 표 3(a)는 연산을 위한 객체를 나타낸다. 표 3(b)는 각 객체에 대한 지시자를 나타낸다. 예를 들어, 'O_X'는 'Mixed MetaOntology'를 나타내고, 'O_{X'}'는 연산 후의 'Mixed MetaOntology'를 나타낸다. 표 3(c)는 각 객체의 포함 관계를 나타낸다. 예를 들어 'E = {C, P, T}'는 '메타온톨로지 요소(E)'가 MetaClass, MetaProperty, MetaRelation으로 구

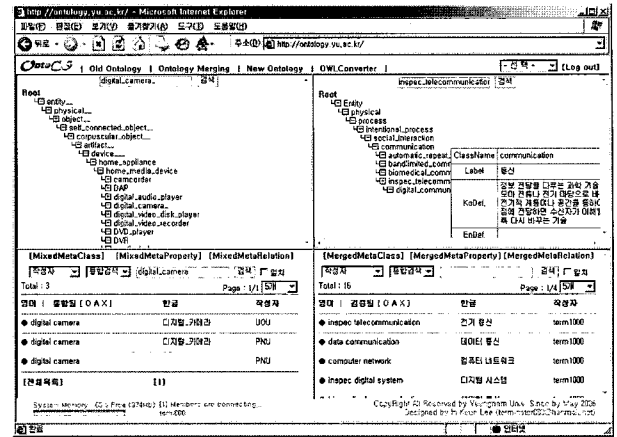


그림 10. OntoIS의 사용자 화면
Fig. 10. User Interface of OntoIS

성되어 있다는 것을 뜻한다. 그리고 표 3(d)는 복사, 생성, 수정, 삭제, 선택, 반복연산과 같이 총 6개의 기본 연산자를 나타낸다. 예를 들어, 'E' = edit(E)는 '메타온톨로지 요소(E)'를 수정(edit())한 결과로써 '연산후의 메타온톨로지 요소(E')'를 출력한다는 의미이다. 그리고 'E' = edit(E, I)는 'E'에서 요소 정보(I)를 수정(edit())한 결과로써 연산 후의 '메타온톨로지 요소(E')'를 출력한다는 의미이다. 그리고 $\left(\bigcup_{k=1}^n E_k\right) = repeat_{k=1}^n(\Omega_k)$ 는 반복연산을 제외한 기본연산 (Ω_k) 을 n회 반복(repeat())하여 연산 결과 집합 $\left(\bigcup_{k=1}^n E_k\right)$ 을 출력한다는 의미이다.

4.3 온톨로지 통합 연산자 조합

'메타온톨로지 요소'의 통합에서 고려할 사항들을 나열하고, 식 (1~6)과 같이 온톨로지 통합 기본 연산자를 조합한 새로운 연산자를 정의한다. 그리고 '메타온톨로지 요소'의 병합과 분리에 대한 상세 과정을 표현한다.

(i) MetaClass의 병합

$$C_M \leftarrow Merge(C_{X_1}, C_{X_2}, \dots, C_{X_n}) \quad (1)$$

◆ 동일 형태의 MetaClass 병합

• MetaClass의 표기와 정보가 같은 경우
 $C_M = choice(C_{X_1}, \dots, C_{X_n}), (I_{C_{X_1}} = \dots = I_{C_{X_n}})$
 $O_M' = add(O_M, C_M)$

• MetaClass의 표기는 같으나 정보가 다른 경우
 $C_M = choice(C_{X_1}, \dots, C_{X_n}), (I_{C_{X_1}} \neq \dots \neq I_{C_{X_n}})$
 $I_{C_M} = choice(I_{C_{X_1}}, I_{C_{X_2}}, \dots, I_{C_{X_n}})$
 $I_{C_M'} = edit(I_{C_M})$
 $C_M' = edit(C_M, I_{C_M'})$
 $O_M' = add(O_M, C_M')$

◆ 다른 형태의 MetaClass 병합

• MetaClass의 표기가 다르나 정보가 같은 경우
 $C_M = choice(C_{X_1}, \dots, C_{X_n}), (I_{C_{X_1}} = \dots = I_{C_{X_n}})$
 $I_{C_M} = choice(I_{C_{X_1}}, I_{C_{X_2}}, \dots, I_{C_{X_n}})$

표 3. 메타온톨로지 통합 연산자
Table 3. Operators for MetaOntology integration
(a)object symbols

O : MetaOntology
E : Element of MetaOntology
C : MetaClass
C^d : Domain MetaClass
C^r : Range MetaClass
P : MetaProperty
R : MetaRelation
I : Information of element
Ω : Equation

(b)indication symbols

Q : Object
I_Q : Specific information
Q_X : Mixed object
Q_M : Merged object
Q_N : n-th object
Q' : Result object of operation

(c)sets of each object

$Q = \{ O, E \}$
$O = \{ E \}$
$E = \{ C, P, R \}$
$C = \{ I_C \}$
$P = \{ I_P \}$
$R = \{ (C^d, P, C^r), I_R \}$
$I = \{ Name, Label, Definition, etc. \}$

(d)operators

복사 : $E_M = copy(E_X)$
생성 : $O' = add(O, E)$
수정 : $E' = edit(E, I)$ $E' = edit(E)$ $O' = edit(O, E)$
삭제 : $O' = remove(O, E)$
선택 : $E = choice\left(\bigcup_{k=1}^n E_k\right)$ or $E = choice(E_1, E_2, \dots, E_n)$
반복연산 : $\left(\bigcup_{k=1}^n E_k\right) = repeat_{k=1}^n(\Omega_k)$

$$C'_M = edit(C_M, I_{C_M})$$

$$O'_M = add(O_M, C'_M)$$

• MetaClass의 표기와 정보가 다른 경우

$$C_M = choice(C_{X_1}, \dots, C_{X_n}), (I_{C_{X_1}} \neq \dots \neq I_{C_{X_n}})$$

$$I_{C_M} = choice(I_{C_{X_1}}, I_{C_{X_2}}, \dots, I_{C_{X_n}})$$

$$I_{C'_M} = edit(I_{C_M})$$

$$C'_M = edit(C_M, I_{C'_M})$$

$$O'_M = add(O_M, C'_M)$$

(ii) MetaClass의 분리

$$(C_{M_1}, C_{M_2}, \dots, C_{M_n}) \Leftarrow Separate(C_X) \quad (2)$$

◆ 다의의 MetaClass를 각 의미별로 재정의

$$\bigcap_{k=1}^n C_{M_k} = repeat_{k=1}^n(copy(C_X)_k)$$

$$\bigcap_{k=1}^n C_{M'_k} = repeat_{k=1}^n(edit(C_{M_k}))$$

$$O'_M = repeat_{k=1}^n(add(O_M, C_{M_k}))$$

(iii) MetaProperty의 병합

$$C_M \Leftarrow Merge(C_{X_1}, C_{X_2}, \dots, C_{X_n}) \quad (3)$$

MetaProperty(P)의 병합과정은 MetaClass(C)의 병합과정과 같다.

(iv) MetaProperty의 분리

$$(C_{M_1}, C_{M_2}, \dots, C_{M_n}) \Leftarrow Separate(C_X) \quad (4)$$

MetaProperty(P)의 분리과정은 MetaClass(C)의 분리과정과 같다.

(iv) MetaRelation의 병합

$$R_M \Leftarrow Merge(R_{X_1}, R_{X_2}, \dots, R_{X_n}) \quad (5)$$

◆ 동일 형태의 MetaRelation 병합

$$R_X = choice(R_{X_1}, R_{X_2}, \dots, R_{X_n})$$

$$(I_{C_{X_1}^d} = \dots = I_{C_{X_n}^d}) \text{ and } (I_{P_{X_1}} = \dots = I_{P_{X_n}})$$

$$\text{and } (I_{C_{X_1}^r} = \dots = I_{C_{X_n}^r})$$

$$R_M = copy(R_X)$$

$$(C_M^d \in O_M, P_M \in O_M, C_M^r \in O_M)$$

$$\text{and } (I_{C_M^d} = I_{C_X^d}, I_{P_M} = I_{P_X}, I_{C_M^r} = I_{C_X^r})$$

$$O'_M = add(O_M, R_M)$$

◆ 다른 형태의 MetaRelation 병합

$$R_X = choice(R_{X_1}, R_{X_2}, \dots, R_{X_n})$$

$$(I_{C_{X_1}^d} \neq \dots \neq I_{C_{X_n}^d}) \text{ and } (I_{P_{X_1}} \neq \dots \neq I_{P_{X_n}})$$

$$\text{and } (I_{C_{X_1}^r} \neq \dots \neq I_{C_{X_n}^r})$$

$$C_M \Leftarrow Merge(C_{X_1}^d, C_{X_2}^d, \dots, C_{X_n}^d)$$

$$P_M \Leftarrow Merge(P_{X_1}, P_{X_2}, \dots, P_{X_n})$$

$$C_M \Leftarrow Merge(C_{X_1}^r, C_{X_2}^r, \dots, C_{X_n}^r)$$

$$R_M = copy(R_X)$$

$$R'_M = edit(R_M)$$

$$(C_M^d \in O_M, P_M \in O_M, C_M^r \in O_M)$$

$$\text{and } (I_{C_M^d} = I_{C_X^d}, I_{P_M} = I_{P_X}, I_{C_M^r} = I_{C_X^r})$$

$$O'_M = add(O_M, R'_M)$$

(v) MetaRelation의 분리

$$(R_{M_1}, R_{M_2}, \dots, R_{M_n}) \Leftarrow Separate(R_X) \quad (6)$$

◆ domain class, property, range class의 분리

$$(C_{X_1}^d, C_{X_2}^d, \dots, C_{X_n}^d) \leftarrow Separate(C_X^d)$$

$$(C_{X_1}^r, C_{X_2}^r, \dots, C_{X_n}^r) \leftarrow Separate(C_X^r)$$

$$(P_{X_1}, P_{X_2}, \dots, P_{X_n}) \leftarrow Separate(P_X)$$

$$\bigcap_{k=1}^n R_{M_k} = repeat_{k=1}^n(copy(R_{M_k}))$$

$$\bigcap_{k=1}^n R_{M_k}' = repeat_{k=1}^n(edit(R_{M_k}))$$

$$O_M' = repeat_{k=1}^n(add(O_M, R_{M_k}))$$

4.4 메타온톨로지 통합 추천 도구

4.4.1 개념 추천

통합될 가능성이 있는 개념의 후보를 제시함에 있어 다음과 같은 방법을 사용한다.

(1) 동일 형태의 개념 제시

개념명(concept name)이 동일한 형태일 경우, 이들은 동일한 의미의 개념일 가능성이 높다. 따라서 동일한 형태의 개념을 제시한다.

(2) 동일 의미의 개념 제시

온톨로지 구축자에 따라 같은 의미의 개념명을 표기할 때, 다른 형태의 단어이지만 같은 의미로 사용되는 단어를 개념명으로 사용하기도 한다. 따라서 소스 온톨로지에서의 동의 개념 관계와 WordNet[11]의 'synset'을 기반으로 동의 개념을 제시한다.

(3) 약어 변환

개념명을 표기할 때 일반적으로 약어를 많이 사용한다. 이런 경우 풀어쓴 형태의 의미와 약어의 의미가 동일하다 하더라도 개념명의 표기는 완전히 달라진다. 따라서 풀어쓴 형태의 개념명을 약어로 변환하여 유사도 측정 시 활용한다.

(4) 유사 형태의 개념 제시

개념을 표기할 때 단·복수를 구분하지 않거나, 복합용어의 경우에는 개념 표기에 약어를 사용함으로써 같은 개념이지만 표기가 조금씩 다를 수 있다. 따라서 개념 표기 간의 유사도를 측정하여 형태적으로 유사한 개념들을 제시한다.

4.4.2 개념의 상·하 관계 추천

특정 개념에 대한 상위 개념(super concept) 후보를 제시한다. 상위 개념 후보를 제시하기 위해 다음과 같은 '계층 구성 규칙'[2]을 정의하여 사용한다.

(1) 중심어 인식

영어 개념명에서의 중심어는 복합용어의 마지막 부분에 위치한다. 단, {by, in, on, of, from, for, with, about}의 전치사로 구분되는 복합용어는 전치사 앞의 용어가 중심어가 된다.

형태) A_B → B(중심어)

예) network_circuit → circuit(중심어)

(2) 동일 중심어 규칙

같은 중심어를 가지면 본질적으로 속성이 같다고 가정한다.

형태) A_B_C is-a B_C(중심어)

예) active_antenna_array is-a antenna_array(중심어)

영어 표현은 다르지만 대역어의 중심어가 같다면 속성이 같다는 가정한다. 아래에서 'C'와 'D'는 한글 대역어가 'c'와 동일한 경우이다.

형태) A_B_C(대역어:a_b_c) is-a B_D(대역어:b_c)

예) personal_digital_assistant(개인 휴대 정보 단말기) is-a terminal(단말기)

(3) 중심어 관계 이행 규칙

영어, 한글 용어 표기는 다르나 개념의 의미가 같은 경우 WordNet의 'synset'을 참조한다.

예) software, program, listing은 같은 중심어로 판단

5. 온톨로지 통합 실험

본 실험에서는 소스 온톨로지를 OntoIntegProcess와 OntoIS를 이용하여 통합하는 과정을 설명한다. 이를 위해 온톨로지 구축 그룹에서 개별적으로 구축한 3개의 온톨로지(O₁, O₂, O₃)를 사용하였다. 그림 11은 각 그룹에서 구축한 온톨로지의 일부를 보이는데, 이들은 개념명과 속성명의 표기 방법이 서로 다르나 개념간의 상·하 관계에 대한 구성은 유사하다.

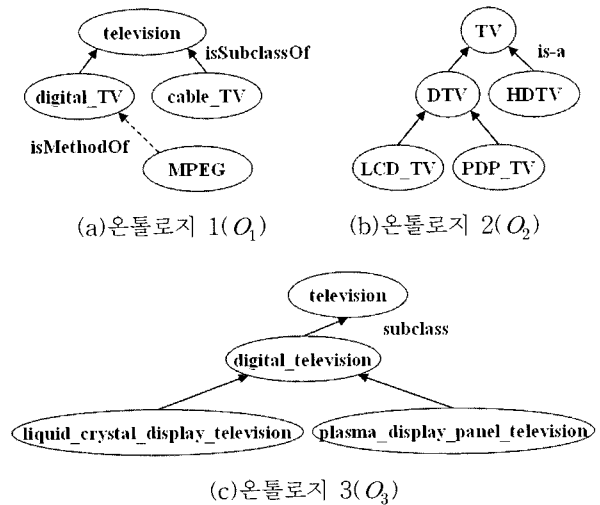
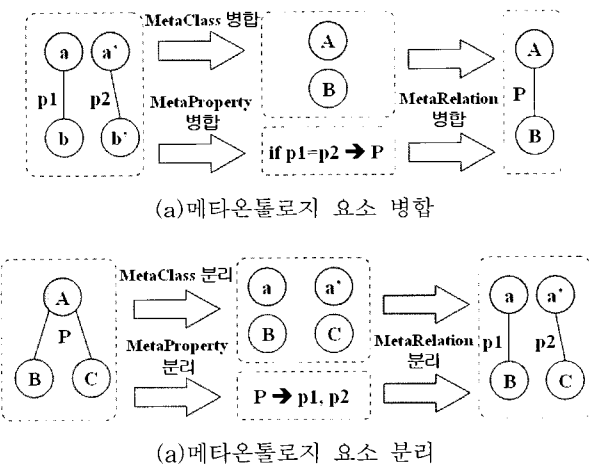


그림 11. 소스 온톨로지의 예

Fig. 11. Examples of the source ontologies



(a) 메타온톨로지 요소 병합

(b) 메타온톨로지 요소 분리

그림 12. 메타온톨로지 요소 통합의 예

Fig. 12. Examples of MetaOntology element integration

온톨로지 통합 과정의 효율을 높이기 위해, 메타온톨로지의 통합은 “MetaClass 통합”, “MetaProperty 통합”, “MetaRelation 통합”의 순서대로 수행하였다. 그림 12는 메타온톨로지 요소의 병합과 분리 과정을 보인다. 그림 12(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 “MetaClass 병합”을 수행하여 개념 ‘A’와 개념 ‘B’를 정의한다. 그리고 속성 ‘p1’과 속성 ‘p2’가 의미적으로 동일하나 표기가 다른 경우, “MetaProperty 병합”을 수행하여 속성 ‘P’를 정의한다. 또한, ‘p1(a,b)’와 ‘p2(a’,b’)'의 관계를 “MetaRelation 병합”을 수행하여 ‘P(A,B)’의 관계를 설정한다. 그림 12(b)는 온톨로지 요소의 분리 과정을 보인다. 즉, 개념 ‘A’의 의미가 온톨로지 설계에 따라 두 개의 개념으로 분리 가능한 경우, 먼저 “MetaClass 분리”를 수행하여 개념 ‘a’와 ‘a’를 정의한다. 그리고 속성

‘P’ 또한 “MetaProperty 분리”를 수행하여 ‘p1’과 ‘p2’를 정의한다. ‘P(A,B)’와 ‘P(A,C)’ 관계를 “MetaRelation 분리”를 수행하여 새롭게 정의한 개념과 속성으로 ‘p1(a,B)’와 ‘p2(a’,C)’ 관계를 설정한다.

표 4는 그림 12의 순서로 소스 온톨로지(O_1, O_2, O_3)를 통합하여 통합된 온톨로지(O_M)을 구축하는 과정을 보인다. 그림 13은 소스 온톨로지를 통합하는 과정에서 OntoIS에서 지원하는 ‘온톨로지 통합 추천 도구’의 사용자 화면을 보인다. 그림 13(a)는 “DTV”와 유사한 개념인 “digital_TV(O_1)”, “DTV(O_2)”, “digital_television(O_3)”을 추천하는 것을 볼 수 있다. 그리고 그림 13(b)는 “digital_television”의 상위 개념으로 “television(O_1)”, “television(O_3)”, “TV(O_2)”를 추천하는 것을 볼 수 있다.

표 4. 메타온톨로지 통합 절차
Table 4. Procedures for MetaOntology integration
(a)MetaClass 통합

Mixed MetaOntology	Operator	Integrated MetaOntology
television(O_1), TV(O_2), television(O_3)	Merge	television(O_M)
digital_TV(O_1), DTV(O_2), digital_television(O_3)	Merge	digital_television(O_M)
cable_TV(O_1)	Merge	cable_television(O_M)
HDTV(O_2)	Merge	high-definition_television(O_M)
LCD_TV(O_2), liquid_crystal_display_television(O_3)	Merge	liquid_crystal_display_television(O_M)
PDP_TV(O_2), plasma_display_panel_television(O_3)	Merge	plasma_display_panel_television(O_M)
MPEG(O_1)	Separate	MPEG_method(O_M), MPEG_group(O_M)

(b)MetaProperty 통합

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

(c)MetaRelation 통합

Mixed MetaOntology	Operator	Integrated MetaOntology
isSubclassOf(digital_TV, television)(O_1) is-a(DTV, TV)(O_2) subclass(digital_television, television)(O_3)	Merge	isSubclassOf(digital_television,television)(O_M)
isSubclassOf(cable_TV,television)(O_1)	Merge	isSubclassOf(cable_television,television)(O_M)
is-a(HDTV, TV)(O_2)	Merge	is-a(high-definition_television,television)(O_M)
is-a(LCD_TV,DTV)(O_2) subclass(liquid_crystal_display_television, digital_television)(O_3)	Merge	isSubclassOf(liquid_crystal_display_television, digital_television)(O_M)
is-a(PDP_TV,DTV)(O_2) subclass(plasma_display_panel_television, digital_television)(O_3)	Merge	isSubclassOf(plasma_display_panel_television, digital_television)(O_M)
isMethodOf(MPEG,digital_TV)(O_1)	Separate	isMethodOf(MPEG_method,digital_television)(O_M)

5개 [DTV] 검색	
동일 형태	● DTV (CNU)
동의 표현	● DTV (CNU) ● digital_television (PNU) ● digital_TV (YNU)
유사 약어	● digital_TV (YNU) ● DTV (CNU) ● LCD_TV (CNU) ● HDTV (CNU) ● PDP_TV (CNU)
포함 형태	● HDTV (CNU)

(a)개념 추천 도구

5개 [digital_television] 검색	
소스 온톨로지 상위 클래스 후보	● television (YNU) ● television_ (PNU) ● TV (CNU)
통합 온톨로지 상위 클래스 후보	● television (MERGE)

(b)개념의 상·하 관계 추천 도구

그림 13. 온톨로지 통합 추천 도구의 사용자 화면

Fig. 13. Recommender's UI on OntoIS

그림 14는 표 4의 온톨로지 통합 과정을 통해 온톨로지를 통합한 결과를 보이며, 그림 15는 그림 14의 온톨로지 통합 결과를 그림으로 나타낸 것이다.

Asserted Hierarchy	Class Property														
<ul style="list-style-type: none"> ● owl:Thing ▶ Entity (5) <ul style="list-style-type: none"> ▶ television (5) <ul style="list-style-type: none"> ● high_definition_television ▶ digital_television (2) <ul style="list-style-type: none"> ● liquid_crystal_display_television ● plasma_display_panel_television ● cable_TV ▶ Temp (2) <ul style="list-style-type: none"> ● MPEG_method ● MPEG_group 	<table border="1"> <tr> <td>EnClass</td> <td>MPEG_method</td> </tr> <tr> <td>KoClass</td> <td>동화상 압축 방식</td> </tr> <tr> <td>KoDef.</td> <td>동화상 압축 방식. 시간에 따라 연속적으로 변화하는 동화상 압축과 코드를 포함할 통해 정보의 전송이 이루어질 수 있는 방법</td> </tr> <tr> <td>EnDef.</td> <td></td> </tr> <tr> <td>Synonym</td> <td></td> </tr> <tr> <td>Writer</td> <td>[이인근]YNU : 2007-02-26 20:25</td> </tr> <tr> <td>Property</td> <td>isMethodOf digital_television</td> </tr> </table>	EnClass	MPEG_method	KoClass	동화상 압축 방식	KoDef.	동화상 압축 방식. 시간에 따라 연속적으로 변화하는 동화상 압축과 코드를 포함할 통해 정보의 전송이 이루어질 수 있는 방법	EnDef.		Synonym		Writer	[이인근]YNU : 2007-02-26 20:25	Property	isMethodOf digital_television
EnClass	MPEG_method														
KoClass	동화상 압축 방식														
KoDef.	동화상 압축 방식. 시간에 따라 연속적으로 변화하는 동화상 압축과 코드를 포함할 통해 정보의 전송이 이루어질 수 있는 방법														
EnDef.															
Synonym															
Writer	[이인근]YNU : 2007-02-26 20:25														
Property	isMethodOf digital_television														
11 Classes	1 Relations														

그림 14. 온톨로지 통합 결과

Fig. 14. Result of ontology integration

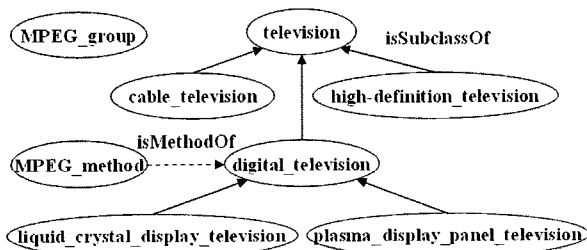


그림 15. 통합된 온톨로지(O_M)

Fig. 15. Integrated ontology(O_M)

6. 결론 및 향후 연구과제

온톨로지의 구축 분야와 사용 목적에 따라 온톨로지를

설계하고 구축하는데 많은 시간과 노력이 필요하다. 따라서 기존에 구축되어 사용되고 있는 온톨로지를 재사용하여 새로운 온톨로지를 구축하려는 연구가 진행되고 있다. 소스 온톨로지와 구축하려는 온톨로지의 설계가 비슷한 경우, 기존의 온톨로지 통합 방법과 도구들은 그 활용도가 높다. 그러나 온톨로지의 설계가 다른 경우에는 온톨로지 통합시 많은 시간과 노력이 소요된다. 또한, 기존의 온톨로지 통합 도구들은 서로 다른 언어로 표현된 온톨로지를 통합하기 어렵다. 따라서 설계와 언어가 다른 다수의 온톨로지를 통합하기 위한 방법이 필요하다.

본 논문에서는 다수의 작업자가 다양한 언어로 표현된 다수의 온톨로지를 통합하기 위한 방법을 제안하였다. 이 방법에서는 복잡한 온톨로지 통합 과정을 간략화하기 위해 온톨로지 요소를 통합하는 개념을 확립하였고, 온톨로지 요소 통합 연산자를 정의하였다. 그리고 온톨로지 요소 통합 연산자를 조합하여 온톨로지 통합에 활용하였다. 또한, 제안한 온톨로지 통합 방법의 구현을 위해 메타온톨로지를 이용한 온톨로지 통합 프로세스를 제안하였다. 그리고 제안한 온톨로지 통합 프로세스에 기반하여 온톨로지 통합 시스템을 개발하였고, 온톨로지 통합 시스템을 이용한 온톨로지를 통합 실험을 통해 제안한 프로세스의 효용성을 확인하였다.

통합한 온톨로지를 사용하기 위해서는 개념, 개체, 속성, 관계와 같은 온톨로지 요소의 정의 이외에도 제약사항과 같은 온톨로지 추론에 필요한 추가적인 정보가 필요하다. 그러나 온톨로지 추론에 필요한 정보를 한 번의 통합 작업으로 온톨로지에 기록하는 것은 어려운 일이다. 즉, 온톨로지 요소의 통합 후, 별도의 작업을 통해 제약사항을 설정해야 한다. 따라서 추론을 위해 필요한 제약사항이나 개념간의 관계 정보를 효율적으로 통합할 수 있는 온톨로지 통합 방법의 연구가 필요하다.

참고 문헌

- [1] 이인근, 서석태, 정혜천, 황도삼, 권순학, “온톨로지 구축 프로세스와 시스템,” *한국퍼지 및 지능 시스템학회 논문지*, 제16권, 6호, pp. 721-729, 2006.
- [2] 황금하 외, “온톨로지 구축 지침서 - BT/NT 관계를 이용한 is-a관계 설정,” CORE-ONTO-118, KAIST, 2006.
- [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. Ponnampereuma, “Dynamic Ontology Integration in a Multi-agent

Environment,” *Proc. of the 20th International Conference on Advanced Information Networking and Applications*, 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] Wordnet, <http://wordnet.princeton.edu>, January 2007.

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

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

저 자 소 개



이인근(In K. Lee)
 2001년 : 영남대학교 재료금속공학부 (공학사)
 2004년 : 영남대학교 대학원 전기공학과 (공학석사)
 2006년~현재 : 동 대학원 전기공학과 박사과정

관심분야 : 지식 기반 지능 시스템, 온톨로지, 영상 이해
 E-mail : iklee@ynu.ac.kr



황도삼(Dosam Hwang)
 1980년 : 홍익대학교 전자계산학과 (이학사)
 1983년 : 연세대학교 전자계산전공 (공학석사)
 1995년 : 교토대학교 전자통신전공 (공학박사)

1980년~1996년 : 한국과학기술원 SERI 책임연구원
 1996년~현재 : 영남대학교 전자정보공학부 교수

관심분야 : 자연언어처리, 온톨로지, 지식처리
 E-mail : dshwang@ynu.ac.kr



서석태(Suk T. Seo)
 2004년 : 영남대학교 전자정보공학부 (공학사)
 2006년 : 영남대학교 대학원 전기공학과 (공학석사)
 2006년~현재 동 대학원 전기공학과 박사과정

관심분야 : 지능 시스템, 영상 처리, 영상 이해
 E-mail : kenneth78@ynu.ac.kr



권순학(Soon H. Kwon)
 1983년 : 서울대학교 제어계측공학과 (공학사)
 1985년 : 서울대학교 대학원 제어계측공학과 (공학석사)
 1995년 : 동경공업대학 시스템과학 (공학박사)
 1996~현재 : 영남대학교 전기공학과 교수

관심분야 : 지식 기반 지능 시스템
 E-mail : shkwon@ynu.ac.kr