

온톨로지 구축 프로세스와 시스템

Ontology Construction Process and System

이인근* 서석태* 정혜천* 황도삼** 권순학*

In K. Lee*, Suk T. Seo*, Hye C. Jeong*, Dosam Hwang**, Soon H. Kwon*

*영남대학교 전기공학과

**영남대학교 전자정보공학부

요 약

컴퓨터를 활용한 지식과 정보 처리를 위해 온톨로지를 구축하고 활용하기 위한 연구가 이루어지고 있다. 그러나 현재까지의 온톨로지 개발 방법 및 온톨로지 구축 도구는 온톨로지 개발 목적에 따라 제한적인 부분에서 연구되어 사용되고 있다. 그러므로 개발하려는 온톨로지 특성에 따른 적절한 온톨로지 구축 프로세스와 도구가 필요하다.

본 논문에서는 특정 분야의 비전문가가 언어 자원으로 부터 지식을 개념화하고, 개념을 형식화하여 온톨로지를 구축할 수 있는 온톨로지 구축 프로세스(OntoProcess¹⁾)를 제안한다. 그리고 다수의 온톨로지 구축자가 동시에 온톨로지 구축 작업을 행할 경우 i)지식의 개념화 과정에서 동일한 개념을 중복 정의하거나 ii)개념의 형식화 과정에서 형식언어와 도구사용법의 이해 부족으로 인해 온톨로지 구축 능률이 저하되는 문제가 발생할 수 있다. 이런 문제를 해결하기 위해 메타 온톨로지를 이용한 다중 온톨로지 구축 프로세스(OntoMProcess²⁾)를 제안한다. 제안한 프로세스에 기반한 온톨로지 구축 시스템(OntoCS³⁾)을 개발하고, 실제 온톨로지 구축 실험을 통하여 제안한 프로세스와 시스템의 효율성을 확인한다. 그리고 온톨로지 구축 과정에서 발견된 문제점 및 이에 대한 해결 방안을 제시한다.

키워드 : 온톨로지, 온톨로지 구축 프로세스, 메타 온톨로지, OntoCS

Abstract

Numbers of research on ontology construction and its application are being done for knowledge and information processing using computers. But, the current ontology development methods and ontology construction tools are using in restricted field on propose. Therefore, proper ontology development processes and ontology construction tools on ontology characteristic are needed.

In this paper, we propose ontology construction process(OntoProcess¹⁾) that non-experts in specific field are able to construct ontology through conceptualization of knowledge and formalization of concepts from language resource. Beside, some problems may be occurred while numbers of people are working together to construct ontology: i)duplicated concept definition in conceptualization process of knowledge and ii)decreasing efficiency of ontology construction by short understanding about formal language and tool operation in formalization process. To solve the problems, we propose an ontology construction process for multiple developers (OntoMProcess²⁾) using meta ontology. We develop an ontology construction system(OntoCS³⁾) based on proposed processes, and we show the efficiency of proposed processes and system from ontology construction experiment.

Key Words : ontology, ontology construction process, meta ontology, OntoCS

1. 서 론

컴퓨터를 활용한 지식 및 정보의 처리를 위해서 온톨로지를 개발하고 활용하려는 연구가 활발히 행해지고 있다. 또한 온톨로지의 개발을 위한 온톨로지 구축 도구들도 다양하게 개발되고 있다. 그러나 분야별로 개발되는 온톨로지의 특성이 다양하여 적절한 온톨로지 개발을 위한 연구 역시 많이 이루어지고 있다. 현재는 온톨로지 개발 이론과 온톨로지 구축 도구가 온톨로지를 개발하는 목적에 따라서 제한적으로 사용되고 있다. 그러므로 적절한 온톨로지 개발 방법 및 온톨로지 구축 도구의 선택은 매우 어려운 일이다.

일반적인 온톨로지의 개발 방법은 목적 확인, 지식의 개념화, 개념의 형식화, 온톨로지 통합, 온톨로지 평가, 문서화의 과정으로 이루어져 있다[4][5]. 지식의 “개념화(conceptualization)”는 어떤 주제에 대해서 핵심적 개념은 무엇이며, 개념간의 관계를 어떻게 설정하는지를 확인하는 과정이다. 즉, 특정 분야(domain)의 지식으로부터 개념을 정의하고 개념 간의 관계를 설정하는 과정으로서 분야 전문가의 개입이 필요한 단계이다. 개념의 “형식화(formalization)”는 자연언어로 작성된 메타 정보로부터 온톨로지의 표현 언어로 기호를 생성하는 과정이다. 즉, 개념과 개념의 관계를

1) Ontology Construction Process의 약자임
2) OntoProcess for Multi Developer의 약자임
3) Ontology Construction System의 약자임

컴퓨터가 이해할 수 있는 정형화된 언어로 표현하는 단계로서 컴퓨터 전문가가 필요한 단계이다. 그러므로 온톨로지 구축 과정에서는 각 분야 전문가의 개입이 필수적이다.

전문가가 충분하지 않은 집단에서의 온톨로지 구축 속도나 범위는 제한적인 수밖에 없고, 비전문가로 구성된 집단에서는 양질의 온톨로지 구축을 기대할 수 없다. 그러나 비전문에게 부족한 분야 전문지식을 보완하기 위해 분야 전문가가 구성한 언어 자원(사전, 코퍼스 등)으로부터 분야 전문지식을 추출하여 온톨로지를 구성한다면 비전문가도 충분히 분야 온톨로지를 구축할 수 있다.

그리고, 다수의 온톨로지 구축자가 같은 분야의 온톨로지를 구축할 경우, 지식의 개념화 과정에서 동일한 개념의 중복이 발생할 수 있으며, 개념의 형식화 과정에서 형식 언어와 온톨로지 구축 도구 사용법에 대한 이해 부족으로 인해 온톨로지 구축 능력이 저하될 수 있다. 이런 문제는 구축 중인 온톨로지를 공유하여 동일한 개념의 중복 구축을 막고, 간단한 형태의 지식 표현 형식을 사용함으로써 복잡한 형식 언어와 도구 사용법에 대한 이해 부족 문제를 해결할 수 있다.

본 논문에서는 특정 분야의 전문가 또는 비전문가가 언어 자원으로부터 지식을 개념화하고 개념을 형식화하여 온톨로지를 구축할 수 있는 온톨로지 구축 프로세스(OntoProcess)를 제안한다. 그리고 다수의 온톨로지 구축자가 동시에 온톨로지 구축 작업을 행할 경우 발생할 수 있는 문제를 해결하기 위해 메타 온톨로지를 이용한 다중 온톨로지 구축 프로세스(OntoMProcess)를 제안한다. 제안한 프로세스에 기반하여 온톨로지 구축 시스템(OntoCS)을 개발하고, 실제 온톨로지 구축 실험을 통하여 제안한 프로세스와 시스템의 효율성을 확인한다. 그리고 온톨로지 구축 과정에서의 문제점 및 문제점에 대한 해결 방안을 제시한다.

2. 기존의 온톨로지 개발 관련 연구

2.1. 기존의 온톨로지 개발 방법

현재까지의 온톨로지 개발 방법은 이론적, 경험적으로 다양하게 제안되어 왔다. 온톨로지 개발 방법으로는 SENSUS, KACTUS, OTK, CommonKADS, Tove, Methontology, Mikrokosmos, ONIONS, PHYSSYS 등 여러 종류가 있다. 그러나 온톨로지 개발의 표준안은 아직 존재하지 않으며, 온톨로지 사용 목적에 따라 적절히 보완되어 사용되고 있다[5].

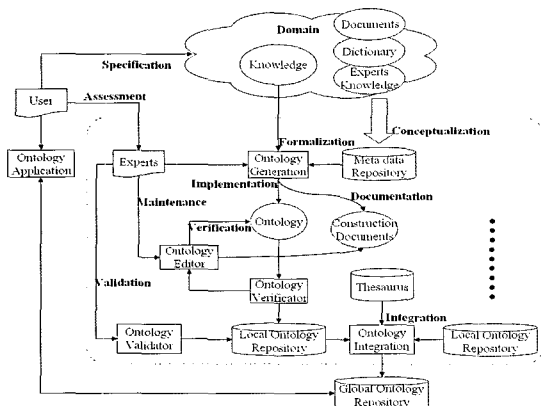


그림 1. 온톨로지 개발 프로세스
Figure 1. Ontology Development Process

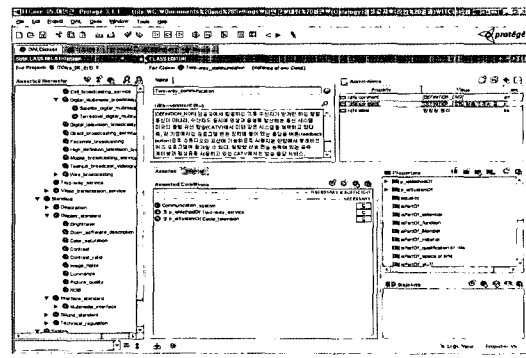


그림 2. Protégé의 사용자 화면
Figure 2. Protégé User Interface

기존 온톨로지 방법론에 근거한 일반적인 온톨로지 개발 프로세스는 그림 1과 같이 지식의 개념화, 개념의 기호화, 온톨로지 검증, 평가, 통합, 문서화와 같은 과정으로 이루어져 있다[1].

2.2. 기존의 온톨로지 구축 도구

온톨로지에 관한 연구가 진행되면서, 온톨로지 구축 도구가 다양하게 개발되었다. 대표적 온톨로지 구축 도구들은 KAON, Protage, OilEd, OWL Editor[11] 등이 있으며, 이들은 쉽게 온톨로지를 구축할 수 있도록 편리한 사용자 인터페이스를 제공한다.

개발된 도구들 중 Protégé는 NSF(National Science Foundation), 국립의료도서관(National Library of Medicine), DARPA(Defense Advanced Research Projects Agency)의 후원을 받아 스탠포드 의과대학의 의료정보학과에서 지식기반의 구조를 작성하기 위한 시스템으로 15년간의 연구 기간을 거쳐 개발되었다. 초기 Protégé는 1988년 경류센(Musen)에 의해 개발되었으며, 1999년 Protégé-2000이 발표되었고, 그 후 Protégé 3.1 (2005), Protégé 3.2 beta(2006)가 지원되고 있으며, 2006년 현재 등록된 사용자만 50,000명이 넘는다[7].

그리고, Protégé는 다양한 분야의 온톨로지를 구축하고, 구축한 온톨로지를 시각화하며, 온톨로지를 관리할 수 있는 환경을 제공한다. 또한, 어떤 특정한 응용 영역에서의 전문가 시스템을 만들기 위해서는 지식 획득이 요구되는데, Java-based Application Programming Interface(API)[8]를 통해 이러한 지식 획득을 지원하기 위한 도구의 개발에 도움을 준다. 그림 2는 Protégé의 사용자 화면을 보인다.

3. 온톨로지 구축 프로세스

3.1. 온토 프로세스(OntoProcess)

그림 1에 제시한 온톨로지 개발 프로세스 중 특정 분야의 언어 자원에 기반하여 온톨로지 구축자가 전문 지식을 개념화하고, 개념화된 지식을 형식화하는 상세 과정으로서 언어 자원에 근거한 온톨로지 구축 프로세스(OntoProcess)를 제안한다.

[온토 프로세스]

[setp 1] 씨앗 용어(seed term)를 키워드로 하여 언어 자원으로부터 특정 분야 관련 문서·문장 검색.

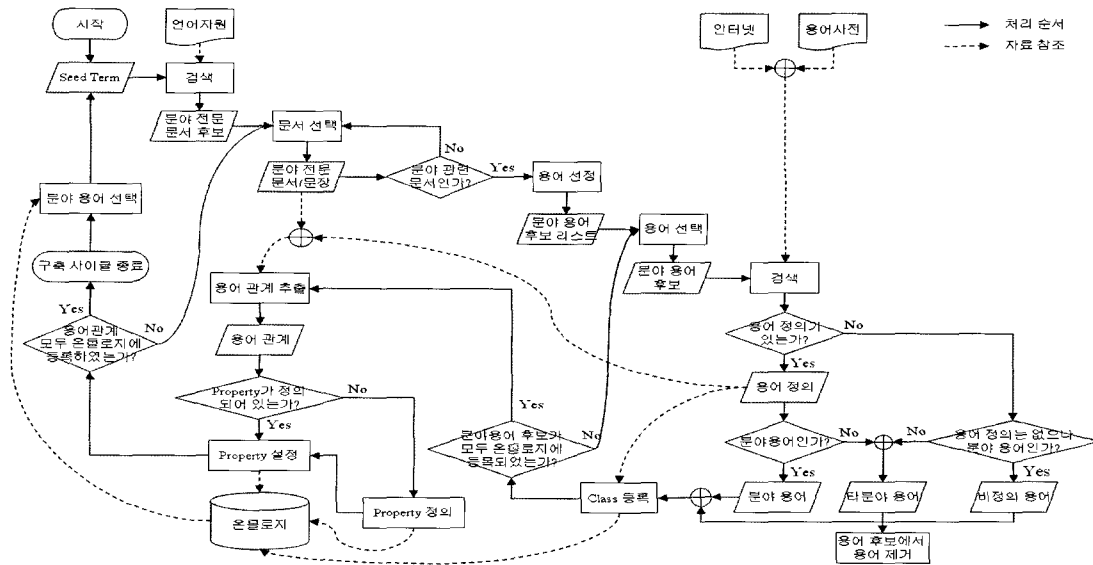


그림 3. 온토 프로세스
Figure 3. OntoProcess

- [setp 2] 문서·문장으로 부터 온톨로지 구축자의 주관에 의해 온톨로지 구축 분야의 분야용어⁴⁾ 후보를 선정.
- [setp 3] 분야용어 후보 중 하나의 용어를 선택.
- [setp 4] 선택된 용어를 온톨로지에서 검색.
- [setp 4-1] 용어가 온톨로지에 등록되어 있으면 [step3] 실행.
- [setp 5] 용어사전과 인터넷을 통해 용어의 정의를 검색.
- [setp 6] 검색된 용어의 정의에 기반 하여 온톨로지 구축자의 주관적 판단에 따라 선택된 용어의 온톨로지 등록 여부를 판단.
- [setp 6-1] 용어의 정의를 알 수 있고 선택한 용어가 분야용어로 판단되면 온톨로지에 등록⁵⁾.
- [setp 6-2] 용어의 정의를 알 수 없으나 선택한 용어가 분야용어로 판단된 경우 비정의 분야용어로 온톨로지에 등록.
- [setp 6-3] 분야용어가 아니라고 판단된 경우 선택한 용어를 후보에서 제거.
- [setp 7] [step2]에서 선정된 분야용어 후보의 온톨로지 등록 여부에 대한 판단이 끝날 때까지 [step2] ~ [step6] 반복.
- [setp 8] [step2]~[step7]에서의 용어의 정의나 문서·문장으로 부터 용어 개념 사이의 관계 추출.
- [setp 8-1] 온톨로지에 관계 정의⁶⁾가 없는 경우 관계 정의를 온톨로지에 추가.
- [setp 8-2] 용어 사이의 관계가 온톨로지에 설정되어 있지 않은 경우 온톨로지에 관계 설정⁷⁾.
- [setp 9] 추출한 용어 관계가 모두 온톨로지에 설정될 때까지 [setp8] 반복

- [setp 10] [step1]~[step9] 반복.
- [setp 10-1] 잘못된 분야용어나 관계는 [step1]~[step9]에서 수정.

그림 3의 온토 프로세스는 사전에서의 용어 표기를 이용하여 개념을 표현하고, 분야별로 분류되어 있지 않은 언어 자원을 이용하여 온톨로지를 구축하는 것을 전제로 한다. 신문 기사, 전문 문서, 사전의 용어 정의 등 분야별로 분류되어 있지 않은 문서에서 관심 문서를 분류하는 많은 연구가 이루어지고 있으나, 새롭게 만들어진 분야 관련 용어가 등록되어 있는 사전이 없이 분야 관련 문서를 분류해 내는 것은 쉬운 일이 아니다. 또한 전문 지식이 부족한 비전문가가 분야 관련 문서를 분류해 내는 것 또한 어려운 일이다. 그러므로 특정 분야를 대표하는 단어를 씨앗 용어(seed term)로 선정하여 언어 자원으로 부터 분야 관련 후보 문서·문장을 검색하고, 검색된 문서들 중 온톨로지 구축자의 주관적 판단으로 적절한 문서를 선택한다.

기존에 제시된 온톨로지 구축 프로세스에서는 온톨로지에 용어를 등록할 때 상위 개념 용어의 등록 유무에 따라 복잡한 용어 등록 절차가 필요하다. 또한 개념간의 관계 설정 단계에서도 전문가에 의해 이미 정의된 관계만을 사용하도록 되어 있다[1]. 그러나 본 논문에서 제안하는 온토 프로세스는 온톨로지 구축 단계에서 온톨로지 구축자가 임의로 관계를 정의하고 개념간의 관계를 설정할 수 있도록 하였다. 그리고 개념의 상·하 관계에 대한 정보가 부족하여 계층을 형성할 수 없다 하더라도, 차후에 정확한 정보를 기반으로 계층을 형성할 수 있도록 하여 온톨로지 구축 능력을 높일 수 있도록 하였다.

3.2. 다중 온토 프로세스(OntoMProcess)

다수의 온톨로지 구축자가 온톨로지를 구축할 때 지식의 개념화 과정에서 동일한 개념을 중복 정의하거나, 개념의 형식화 과정에서 형식 언어와 도구 사용법의 이해 부족으로 인해 온톨로지의 구축 능력이 저하되는 문제가 발생할 수 있다. 다수의 온톨로지 구축자에 의한 동일 개념의 중복 정의

4) 온톨로지에 등록된 용어를 “분야용어”로 표현한다.
 5) 본 논문에서는 사전 용어 정의를 기반으로 개념을 정의하고 사전 용어으로써 개념을 표기한다. 따라서 “개념을 정의”하는 것을 온톨로지에 “용어를 등록”하는 것으로 표현한다.
 6) 두 개념의 관계는 domain_concept-predicate-range_concept 으로 표현할 수 있다. 두 개념의 관계를 표현하기 위해 온톨로지에서 predicate를 정의하는 것을 “관계 정의”라 한다.
 7) 두 개의 분야용어를 정의된 관계로 연결하는 것을 “관계 설정”이라 한다.

문제는 구축 중인 온톨로지와 언어 자원을 온톨로지 구축자들이 네트워크를 통해 공유함으로써 해결할 수 있다. 그리고 복잡한 형식언어를 대신하여 정형화된 테이블 형태로 온톨로지를 표현함으로써 온톨로지 구축자가 쉽게 온톨로지를 구축할 수 있도록 한다.

온톨로지를 표 1과 같이 테이블 형태의 메타 온톨로지(meta ontology)로 표현하고, 메타 온톨로지를 네트워크로 공유하여 온톨로지를 구축하는 다중 온톨로지 구축 프로세스(OntoMProcess)를 제안한다.

표 1(a)는 개념을 나타내는 클래스(class)의 형식이다. 표 1(b)는 개념간의 관계를 나타내는 domain_class-property-range_class의 triple 형식을 나타낸다. <표 1(c)>는 개념의 구체물인 인스턴스(instance)이다. 클래스는 클래스와 관계를 설정할 수 있으나, 인스턴스는 다른 인스턴스(instance object)나 자료형(data type)에 대해서 관계를 설정할 수 있다. 표 1(d)와 표 1(e)는 인스턴스의 관계를 나타낸다. 그리고 온톨로지 구축자가 메타 온톨로지에 관계를 정의할 수 있도록 하였고, 관계 정의 형식은 표 1(f)과 같다.

표 1. 메타 온톨로지 형식과 예
Table 1. Meta ontology form and examples

(a) 클래스 형식 (a) class form	
Class Name	digital television
Korean	디지털 TV
Definition	후백시대 ·컬러시대를 거친, 이른바 제3세대 텔레비전이다.
Synonym	디지털 텔레비전, digital TV

(b) 클래스 관계 설정 형식 (b) class relation form	
Domain	digital television (디지털 TV)
Property	isSubclassOf (이다)
Restriction	someValuesFrom
Range	television (텔레비전)
Source	디지털 TV는 후백시대 ·컬러시대를 거친, 이른바 제3세대 텔레비전이다

(c) 인스턴스 형식 (c) instance form	
Instance	Pavv SVP-42Q2HL1
SuperClass	digital television

(d) 인스턴스 관계 설정 형식 (object) (d) instance relation form of object	
Domain	삼성 (Samsung)
Property	isProducerOf
Range	Pavv SVP-42Q2HL1
Source	http://blog.naver.com/ecoms?Redirect=Log&logNo=80027474383

(e) 인스턴스 관계 설정 형식(data type) (e) instance relation form of data type	
Domain	Pavv SVP-42Q2HL1
Property	panelSize
Data value	42inch
Data type	integer
Source	http://blog.naver.com/ecoms?Redirect=Log&logNo=80027474383

(f) 관계 정의 형식
(e) relation definition form

Property	isProducerOf
Definition	객체들을 조합하여 다른 객체를 만들어 내는 객체
Predicate	제조하다, 생산하다

메타 온톨로지를 이용한 다중 온톨로지 구축 프로세스는 그림 4(b)와 같다. 그림 4(a)와 같이 다수의 온톨로지 구축자가 개별적으로 온톨로지를 구축할 경우, 각 온톨로지 구축자가 구축한 온톨로지를 하나로 통합해야 한다. Prompt[10]와 같은 온톨로지 통합(merging) 도구를 이용하여 온톨로지를 통합할 수도 있으나, 다수의 온톨로지를 하나의 온톨로지 통합 과정은 단순하지 않다. 그러나 다수의 온톨로지 구축자가 메타 온톨로지를 네트워크로 공유하고 온톨로지를 구축하면, 온톨로지의 중복 구축을 사전에 방지할 수 있다. 그리고, OWL 자동 변환기를 이용하여 구축된 메타 온톨로지를 OWL 형태의 온톨로지 자동 변환할 수 있다.

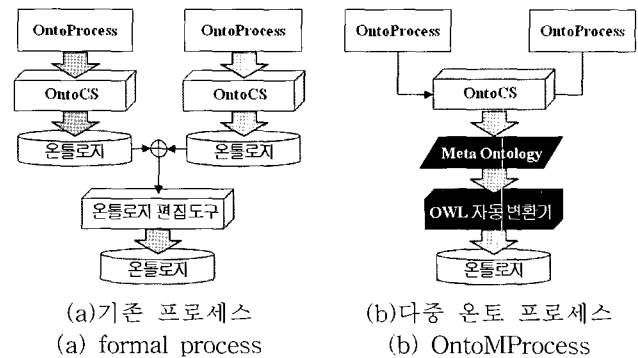


그림 4. 온톨로지 구축 프로세스 비교
Figure 4. Comparison between two ontology construction processes

3.3. 온톨로지 구축 문제 및 해결방법

온톨로지 구축 과정에서 흔히 발생할 수 있는 문제가 개념의 정의 문제이다. 다중 온톨로지 프로세스에 의해 자원을 공유하여 동일한 개념의 중복 구축을 최대한 방지할 수 있으나, 다의적 개념에 대한 개념 표기(class name)에 대한 애매성은 존재한다. 그리고 다수의 온톨로지 구축 그룹에서 발생하는 동일 개념 표기 충돌 문제는 흔히 발생하는 온톨로지 구축 과정에서의 문제라고 할 수 있다. 아래 온톨로지 구축 과정에서 부각된 클래스 표기와 관계 정의, 그리고 계층 구조 형성에서의 문제점을 나열하고 각 문제점에 대한 해결 방법을 제시한다.

3.3.1. 클래스(class) 표기 문제

(1) 대·소문자 구분

OWL[9]은 영어의 대·소문자를 구분한다. 그러므로 같은 철자의 클래스라 하더라도 대·소문자가 일치하지 않으면 서로 다른 클래스로 인식해 버린다. 이 문제는 온톨로지 구축 과정에서 흔히 발생하는 문제이다.

예) Digital_Television ↔ digital_television

대·소문자 구분 문제를 해결하기 위해서는 온톨로지 구축자들 사이에 약속이 중요하며, 특수한 경우를 제외하고 모

든 클래스를 소문자로 표기함을 원칙으로 한다.

(2) 띄어쓰기 구분

복합어의 경우 동일 개념이고 동일 철자이지만 띄어쓰기 방법에 따라 서로 다른 클래스로 인식한다. 이 문제로 인한 클래스의 중복 등록이 많이 발생한다.

예) biotechnology ⇔ bio_technology

띄어쓰기 문제를 해결하기 위해서는 클래스 등록 시 온톨로지 구축자가 온톨로지에 등록된 클래스를 최대한 참조하여 클래스의 중복 등록을 방지해야 한다.

(3) 약어 표기

사전적 표기와 일상용어 사이에 많이 발생하는 문제 중 하나가 약어의 표현이다. 약어는 원 개념 표기와 그 형태가 달라 온톨로지 구축자 사이에서도 중복 등록 여부를 확인하지 못하는 상황이다

예) LCD ⇔ liquid_crystal_displays

동일한 개념을 표현할 때는 약어의 표현은 자제한다. 단, 약어는 클래스 등록 시 동의어로 등록한다.

(4) 단·복수형 표기

사전적 표기와 일상용어 사이에서의 문제 중 또 다른 하나는 단·복수형 표기이다. 사전에는 일반적으로 단수형으로 표현되어 있으나, 온톨로지 구축자가 개념적으로 클래스를 정의할 경우 복수형이 될 가능성이 높다.

예) amplifiers ⇔ amplifier

이 문제는 특별한 경우를 제외하고는 모든 클래스를 단수형으로 표현한다.

(5) 특수문자의 사용

일상적으로는 특수 문자의 사용이 빈번하다. OWL로 온톨로지를 구성할 때 특수 문자를 사용할 경우 인코딩(encoding)이나 디코딩(decoding) 과정에서 오류가 발생할 수 있다.

예) C#, C++

이 문제는 “C#”→“C_sharp”, “C++”→“C_plus_plus”와 같이 특수 문자를 영어 표기로 바꾸어 기록한다.

(6) 다의어 표현

서로 다른 개념이지만 동일한 철자로 사용하는 경우가 있다. 이 용어들은 실제적으로도 혼용하여 사용된다. 이 용어의 의미는 문맥을 통해서만 해석이 가능하다.

예) (1) MPEG(Moving Picture Experts Group) - 동영상 부호화하는 방법을 이용해서 표준을 정립하는 것을 목적으로 하는 동화상 전문가 그룹.

(2) MPEG(Moving Picture Experts Group) - 동화상의 압축 방식

이 문제는 (1)→“MPEG_group”, (2)→“MPEG_method”와 같이 클래스를 표기할 때 상위 개념을 함께 표기한다. 상위 개념이 동일할 경우에는 “MPEG_group_1”과 같이 상위 개념과 번호를 기입한다.

3.3.2. 개념 관계 정의 문제

온토 프로세스에서는 개념 관계를 자유롭게 정의할 수 있다. 이 과정에서 온톨로지 구축자들이 기존에 정의된 개념 관계를 활용하기는 하지만, 작은 의미적 차이로 인해 유사한 개념 관계를 추가 정의하는 경우가 있다.

예) isUsedAs: ~로 사용되다

isUsedIn: 다른 개체 안에서 사용되다

isUsedTo: 다른 개체에 어떤 목적으로 사용
use: 개체의 사용 관계

이 문제의 해결 방법으로는 전문가에 의한 개념 관계의 표준화가 필요하다. 그러므로 표준화 과정에서 개념 관계의 계층을 구성한다.

예) use ⊇ isUsedAs

⊇ isUsedIn

⊇ isUsedTo

3.3.3. 비 구조화된 계층 구조

분야 비전문가 그룹에서 온톨로지를 구축하는 과정에서 전문 지식의 부족으로 인해 클래스의 상·하 관계 설정을 할 때 클래스의 사전적 정의에 의존할 수밖에 없다. 그러나 클래스의 사전적 정의의 표현이 너무 포괄적이어서 세분화된 계층 구성이 쉽지 않다. 고선명 텔레비전을 예로 들면, 고선명 텔레비전의 상식적 계층 관계는 그림 5(a)와 같다. 그러나 사전적 정의를 기반으로 계층 관계를 구성하면 그림 5(b)와 같다.

이 문제는 전문가의 전문 지식이 반영 되어야 하며, 검증 과정에서 전문가가 의해 계층 관계를 구조화한다.

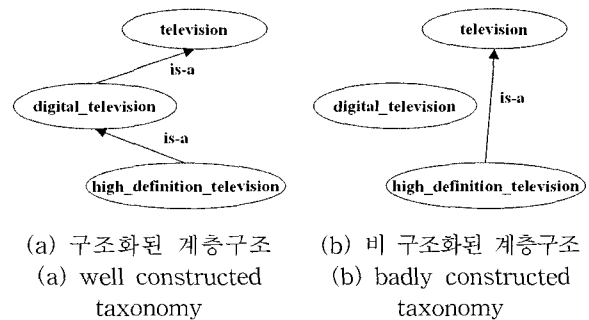


그림 5. 클래스 계층구조

Figure 5. Class Taxonomy

4. 온톨로지 구축 시스템

제안한 온톨로지 구축 프로세스에 따라 온톨로지를 구축하기 위해서는 효율적인 온톨로지 구축 도구가 필요하다. 비전문가가 온톨로지를 구축하기 위해서는 전문 문서 및 사진을 이용해야 한다. 이러한 다수의 언어 자원을 동시에 이용하기 위해서는 다중 작업 공간이 필요하다. 그리고 온톨로지 구축자가 참조할 언어 자원과 구축중인 온톨로지를 공유하기 위한 시스템이 필요하다.

본 논문에서는 제안한 온톨로지 구축 프로세스에 기반하여 i)다수의 온톨로지 구축자가 언어 자원과 메타 온톨로지를 공유하여 ii)언어 자원을 편리하게 활용하고 관리할 수 있고, iii)메타 온톨로지를 효율적으로 관리할 수 있으며, iv)작성한 메타 온톨로지를 OWL형태의 온톨로지로 자동 변환하여 온톨로지 구축을 지원하는 온톨로지 구축 시스템(OntoCS)을 개발한다.

4.1. OntoCS 구조

그림 6은 OntoCS의 구조를 보인다. “코퍼스 자원 관리 도

구”와 “용어 사전 관리 도구”를 이용하여 온톨로지 구축에 필요한 언어 자원을 관리한다. 그리고 “자원 사용 기록(log) 및 히스토리(history) 관리 도구”에서는 각 온톨로지 구축자가 개별적으로 사용한 언어 자원에 대한 목록을 관리할 수 있다. 개념화한 지식을 “메타 온톨로지 관리 도구”를 이용하여 다수의 온톨로지 구축자가 온톨로지 구축 전단계의 “메타 온톨로지”를 작성할 수 있으며, “분야용어 태깅 도구”는 이미 구축된 메타 온톨로지를 참조하여 언어 자원의 문장에서 분야용어에 색상과 링크를 추가하여 관련 용어를 쉽게 선정하고 검색할 수 있도록 한다. 그리고 구축한 메타 온톨로지는 “OWL 변환 도구”를 이용하여 OWL 형태의 온톨로지로 자동 변환될 수 있다. 그림 7은 OntoCS의 사용자 화면이다.

4.2. 온톨로지 구축 관리 시스템

온톨로지를 구축하기 위해서는 온톨로지 구축 목표에 따라 온톨로지가 설계되고, 다수의 온톨로지 구축자로 그룹을 구성하여 설계에 따라 온톨로지가 구축되어야 한다. 온톨로지 구축 과정을 관리하기 위해서는 온톨로지 구축 과정에서 온톨로지 구축 진행 상황과 온톨로지 구축자의 작업 능력에 대해 파악되어야 한다. 따라서 OntoCS와 연동한 온톨로지 구축 관리 시스템을 개발하고 온톨로지 구축 과정을 관리할 수 있도록 하였다.

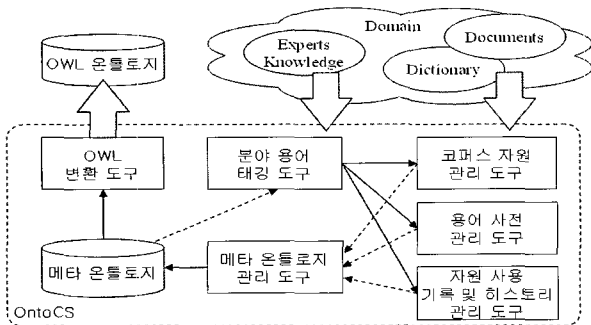


그림 6. OntoCS의 구조
Figure 6. Structure of OntoCS

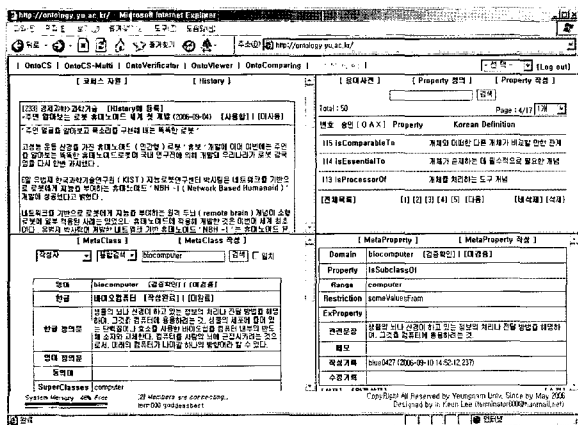


그림 7. OntoCS 사용자 화면
Figure 7. OntoCS User Interface

그림 8은 각 온톨로지 구축자의 일별 온톨로지 구축량에 대해 기록한 것이며, 그림 9는 각 온톨로지 구축자의 일별 온톨로지 구축 작업 시간을 측정하는 것이다.

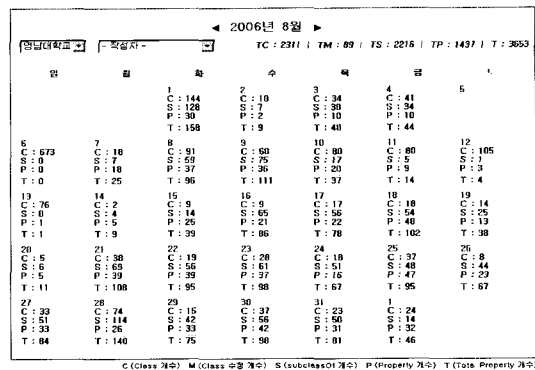


그림 8. 온톨로지 구축량 통계
Figure 8. Daily statistics of ontology

5. 온톨로지 구축 실험

제안한 프로세스의 효율성을 확인 실험을 위해 두 개의 온톨로지 구축 그룹(A와 B)을 구성하였다. A그룹에서는 제안한 프로세스에 기반 하여 온톨로지를 구축하고, B그룹에서는 기존 온톨로지 구축 프로세스에 기반 하여 온톨로지를 구축하였다. 그리고 각 그룹의 온톨로지 구축 결과를 분석하여 제안한 프로세스의 효율성을 확인한다. 실험에서는 표 1(a)의 클래스를 정의하고 표 1(b)의 클래스 관계를 설정하는 것으로 메타 온톨로지를 구축하였다. 클래스와 인스턴스의 구분은 온톨로지의 활용 목적에 따라 적절한 수준에서 결정될 사항이다.

5.1. 온톨로지 프로세스에 기반한 온톨로지 구축 실험

OntoCS를 사용하여 “디지털 TV 및 방송” 분야의 온톨로지를 구축하였다. 구축하려는 온톨로지 분야와 무관한 전공의 대학생 17명으로 온톨로지 구축 A그룹을 구성하였다. 온톨로지 구축자에게는 온톨로지의 개념과 온톨로지 구축 프로세스, OntoCS의 사용법에 대한 기본 교육을 실시하고, 각 개인별로 한 달 동안 20회~30회의 온톨로지 구축 연습을 하도록 하였다.

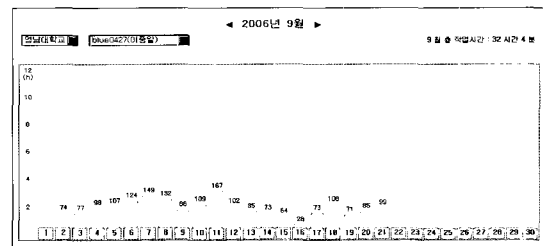


그림 9. 온톨로지 구축시간 통계
Figure 9. Daily statistics of working time

실험에 앞서 A그룹에서 5명을 피실험자로 선별하고, 10회의 온톨로지 구축 실험에서 피실험자의 온톨로지 구축량과 온톨로지 구축 소요시간을 측정하고 분석하였다. 온톨로지 구축량과 온톨로지 구축 소요시간은 그림 8과 그림 9의 온톨로지 구축 관리 시스템의 통계 자료를 참조하였다. 표 2는 실험 종료 후 온톨로지 총 구축량과 전체 실험시간을 보인다. 그림 10은 온톨로지 구축자의 각 실험 회차별 클래스와 클래스 관계의 평균 등록시간을 그래프로 보인 것이다.

표 2. 온톨로지 구축 소요시간 측정 결과
Table 2. Time required for constructing ontology

피실험자	A	B	C	D	E
구축개수(개)	505	564	814	674	1066
작업시간(분)	1122	786	1061	1134	1451
평균등록시간(분)	2.2	1.4	1.3	1.6	1.3

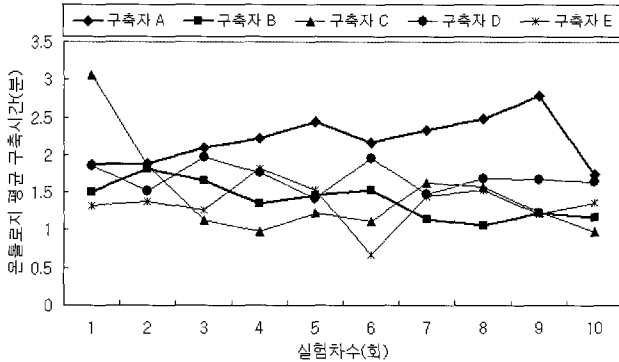


그림 10. 온톨로지 구축 소요시간 측정 그래프
Figure 10. Graph of time for constructing ontology

본 실험 결과는 온톨로지에 대한 이해와 도구 사용법을 숙지한 이후 실시한 실험이므로 실험 회차를 거듭하면서 작업 속도가 빨라졌다고 판단하기는 어렵다. 전반적으로 온톨로지 평균 등록 시간이 단축되기는 하였으나, 피실험자 A의 경우에는 오히려 증가한 경우를 볼 수 있었다. 따라서 5명의 피실험자의 실험결과 중 상반되는 두 개의 실험결과를 표 3과 표 4에 보이고 그 차이점을 분석한다.

표 3. 온톨로지 구축 실험결과 (피실험자 A)
Table 3. Test Result of Subject A

차수(회)	클래스(개)	관계설정(개)	등록비율(클래스:관계설정)	작업시간(분)	평균등록시간(분)
1	9	32	1 : 3.5	77	1.8
2	17	35	1 : 2.0	98	1.8
3	13	38	1 : 2.9	107	2.0
4	21	35	1 : 1.6	124	2.2
5	24	37	1 : 1.5	149	2.4
6	22	39	1 : 1.7	132	2.1
7	9	28	1 : 3.1	86	2.3
8	15	29	1 : 1.9	109	2.4
9	22	38	1 : 1.7	167	2.7
10	10	32	1 : 3.2	73	1.7

두 피실험자의 실험 결과를 볼 때, 피실험자 A의 경우는 실험 회차를 거듭할수록 클래스의 등록 비중이 높아진 반면, 피실험자 B의 경우는 클래스의 관계 설정 비중이 높아진 것을 알 수 있다. 클래스 관계 설정 비중이 높아짐에 따라 평균 등록 시간이 단축되는 원인은 피실험자의 인터뷰를 통해 알 수 있었다. 즉, 표 1에서와 같이 클래스를 등록하기 위해서는 많은 정보를 입력해야 하지만 클래스 관계 설정은 이미 등록된 클래스를 참조만 하므로 클래스 관계 설정 시간이 클래스 등록 시간보다 더 짧다. 그리고 온톨로지로 표현할 수 있는 개념의 수가 제한되어 있으므로 실험이 진행됨에 따라 등록할 클래스의 수가 줄어드는 반면, 이미 등록된 클래스를 참조하는 클래스 관계 설정 개수는 늘어난다는 것을 알 수 있었다.

표 4. 온톨로지 구축 실험결과 (피실험자 B)
Table 4. Test Result of Subject B

차수(회)	클래스(개)	관계설정(개)	등록비율(클래스:관계설정)	작업시간(분)	평균등록시간(분)
1	26	38	1 : 1.4	96	1.5
2	15	49	1 : 3.2	116	1.8
3	5	45	1 : 9.0	83	1.6
4	13	52	1 : 4.0	88	1.3
5	8	47	1 : 5.8	80	1.4
6	7	35	1 : 5.0	64	1.5
7	3	53	1 : 17.6	64	1.1
8	4	44	1 : 11.0	51	1.0
9	1	51	1 : 51.0	64	1.2
10	23	45	1 : 1.9	80	1.1

5.2. 온톨로지 구축결과 비교

제안한 다중 온토 프로세스의 타당성을 검증하기 위해 온톨로지 구축 B그룹을 구성하고, 기존의 온톨로지 구축 프로세스에 기반 하여 온톨로지를 구축하였다. 4명의 온톨로지 구축자로 온톨로지 구축 그룹을 구성하고, Protégé를 사용하여 각기 다른 방법으로 온톨로지를 구축하도록 하였다. 실험에서 구축한 온톨로지는 클래스간의 계층구조와 클래스만으로 구성되었다. 온톨로지 구축 결과 4개의 OWL 파일이 생성되었다. 표 5와 표 6은 B그룹에서 구축한 온톨로지를 분석한 결과이다.

표 5에서도 알 수 있듯이 B그룹에서 구축한 온톨로지에는 다수의 중복되는 클래스가 존재하였다. 그리고 표 6에서와 같이 두 명의 온톨로지 구축자가 등록한 동일한 클래스가 31개이고, 세 명의 온톨로지 구축자 등록한 동일한 클래스는 2개이다.

이 온톨로지는 클래스의 상·하 관계만으로 구성되었으나, 온톨로지를 완성하기 위해서는 클래스 상·하 관계 이외의 관계도 설정해야 한다. 클래스 관계를 설정하기 위해서는 새로운 클래스를 계속 추가 등록해야 하는데, 이 과정에서 심한 클래스의 중복 등록이 예상된다.

비교 실험을 통해 메타 온톨로지를 이용한 다중 온토 프로세스가 클래스의 중복 등록을 방지하고, 다른 구축자에 의해 등록된 클래스를 재사용함으로써 온톨로지 구축 능력이 향상됨을 알 수 있다.

표 5. B그룹 온톨로지의 분석 결과
Table 5. An analysis result of B group's ontology

구축자	A'	B'	C'	D'
구축개수	493	480	60	696
중복개수	20	23	4	21

표 6. B그룹 온톨로지의 중복 클래스 개수
Table 6. Same class counts of B group's ontology

전체 중복회수	0번	1번	2번
계	1661	31	2

6. 결론 및 향후 연구과제

본 논문은 특정 분야의 언어 자원에 근거하여 온톨로지를 구축하는 온톨로지 구축 프로세스(OntoProcess)를 제안하였다. 그리고 다수의 온톨로지 구축자가 동시에 온톨로지 구축

할 경우 발생할 수 있는 문제를 해결하기 위해 메타 온톨로지를 이용한 다중 온톨로지 구축 프로세스(OntoMProcess)를 제안하였다. 또한 제안한 프로세스를 이용하여 비전문가도 쉽게 사용할 수 있는 온톨로지 구축 시스템을 개발(OntoCS)하였다. 개발한 시스템을 이용한 온톨로지 구축 실험을 통해 온톨로지 구축 프로세스와 온톨로지 구축 시스템의 효율성을 확인하였고, 실험 과정에서 온톨로지 구축시 발생할 수 있는 문제점과 해결 방안을 제시하였다.

본 논문에서 제안한 온톨로지 구축 프로세스는 온톨로지 개발 과정에서 지식을 개념화하고, 개념을 형식화 하는 단계에서 활용될 수 있다. 제안한 프로세스에 의해 구축한 온톨로지는 상위 온톨로지로 활용될 수 있으나, 실제 응용 분야에 사용되기 위해서는 온톨로지 개발 계획 단계에서 온톨로지의 사용 목적과 방법 등 구체적인 온톨로지 설계가 이루어져야 한다. 또한, 그 설계에 따라 온톨로지 구축 범위와 수준이 정해지며, 온톨로지 검사 및 검증 과정에서 온톨로지가 수정되어야 할 것이다. 그러므로 완전한 온톨로지를 개발하기 위한 온톨로지 검증, 평가, 통합에 관한 구체적인 연구도 필요하다.

참 고 문 헌

- [1] 이인근, 황도삼, 권순학, "씨앗 용어 피드백 관련 검색에 근거한 온톨로지 구축," 제18회 한글 및 한국어 정보처리 학술대회, pp.81-88, 2006.
- [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, 2001.
- [3] Alan Rector, et al., "OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns," 14th International Conference on Knowledge Engineering and Knowledge Management, Whittlebury Hall, Northamptonshire, UK, 2004.
- [4] "웹 온톨로지 개발지침 연구," 한국전산원, NCA IV-RER-04059, 2004.
- [5] 김은경, 남영준, "시맨틱 웹을 위한 온톨로지 구축 방법에 관한 비교 연구," 정보관리연구, vol.35, no.2, pp.57-85, 2004.
- [6] Hai Wang, Matthew Horridge, Alan L. Rector, Nick Drummond, Julian Seidenberg, "Debugging OWL-DL Ontologies: A Heuristic Approach," International Semantic Web Conference 2005, pp.745-757, 2005.
- [7] Protégé, <http://protege.stanford.edu>, July 2006.
- [8] Protégé OWL API Programmer's Guide, <http://protege.stanford.edu/plugins/owl/api/guide.html>, July 2006.
- [9] OWL Web Ontology Language Guide, <http://www.w3.org/TR/owl-guide>, July 2006.
- [10] Prompt plugin, <http://protege.stanford.edu/plugins/prompt/prompt.html>, July 2006.
- [11] Michael Denny, "Ontology editor survey results," http://xml.com/2002/11/06/Ontology_Editor_Survey.html, June 2002.

저 자 소 개



이인근(In K. Lee)

2001년 영남대학교 재료금속공학부 (공학사)

2004년 영남대학교 대학원 전기공학과 (공학석사)

2006년~현재 동 대학원 전기공학과 박사과정

관심분야 : 지능 시스템, 자연언어처리, 정보검색, 온톨로지, 영상 이해

E-mail : iklee@ynu.ac.kr



서석태(Suk Tae 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



정혜천(Hye C. Jung)

1998년 : 경일대학교 전자공학과 (공학사)

2000년 : 영남대학교 대학원 전기공학과 (공학석사)

2006년~현재 동 대학원 전기공학과 박사과정

2000년~2005년 LG전자 디스플레이 제품 연구소
2005년~현재 한국섬유기계연구소 자동화 기술 팀장

관심분야 : 제어 시스템, 지식기반 지능 시스템

E-mail : doldory@kotmi.re.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@yu.ac.kr



권순학(Soon H. Kwon)

1983년 : 서울대학교 제어계측공학과
(공학사)

1985년 : 서울대학교 대학원 제어계측
공학과 (공학석사)

1995년 : 동경공업대학 시스템과학
(공학박사)

1996~현재 : 영남대학교 전기공학과 교수

관심분야 : 지식 기반 지능 시스템

E-mail : shkwon@yu.ac.kr