OWL 기반 그래픽 바이오 온톨로지 관리 시스템의 설계 및 구현

(Design and Implementation of a Graphical Bio-Ontology Management System based on OWL)

김 기 헌 † 최 재 훈 †† 양 재 동 ††† 박 천 수 ††††
(Ki-Heon Kim) (Jae-Hun Choi) (Jae-Dong Yang) (Cheon-Shu Park)

요 약 본 논문에서는 OWL(Web Ontology Language) 기반 그래픽 바이오 온톨로지 관리 시스템을 설계하고 구현하였다. 이 시스템은 생물학 용어들 사이의 복잡한 의미 관계들로 구성되는 바이오 온톨로지를 본 논문에서 정의한 그래픽 표기를 이용하여 표현한다. 또한, 시각화된 환경에서 수행되는 상속과 역상속 메커니즘은 이미 구축된 방대한 용어들 사이의 관계를 시스템이 구조적으로 파악할 수 있게 함으로써, 전문가가 새로 추가되는 용어에 대한 관계를 의미적으로 일관성 있게 반자동으로 구축할 수 있다. 구축된 온톨로지는 기본적으로 OWL로 기술되며, 다른 여러 표준 온톨로지 언어(RDF/RDFS, DAML+OIL 등)로 의미적 손실 없이 변환된다. 본 시스템의 중요한 특징은 OWL의 강력한 의미적 표현력과 이를 잘 정의할 수 있는 그래픽 표기법을 채택함으로써 시각화된 메커니즘을 통해 바이오 온톨로지를 정교하게 모델링할 수 있다는 점이다.

키워드 : 온톨로지, OWL, 바이오인포매틱스, 지식표현

Abstract In this paper, we design and implement a graphical bio-ontology management system based on OWL(Web Ontology Language). It allows domain experts to easily manage sophisticated bio-ontologies in which biological knowledge is encoded. The knowledge can be seamlessly modeled into the ontology by well defined graphical notations, which capture most of subtle semantics inherently existing between biological terms. Our system provides a new construction mechanism, which can determine a considerable part of relationships between terms by their inheritance and inverse-inheritance. For keeping their semantics to be consistent, the mechanism supplies domain experts with information available from relationships being constructed or already constructed. The constructed ontology is basically formatted by OWL, which may benefit from its powerful semantic expressiveness. Additionally, it can be automatically translated into other standard languages without semantic loss, such as RDF/RDFS, DAML+OIL and so on. The main characteristics of our system is that it enables domain experts to delicately model the bio-ontology by the visualized construction mechanisms adopting well-defined graphical notations based on OWL.

Key words : Ontology, OWL, Bioinformatics, Knowledge Representation

1. 서 론

최근, 생명 관련 분야의 급격한 발전으로 인해 방대한 생명정보가 생성되고 있다. 또한 이 정보를 분석하여 여러 다양한 응용 분야에 이용하기 위해 생명 데이타베이스(DB)를 구축하고 이를 웹서비스를 통해 공유하기 위한 많은 연구들이 진행 중이다. 이러한 DB들의 효율적 분석과 공유를 위해서는 DB들에 내재되어 있는 도메인 지식이 필요하고, 이를 관리하기 위한 한 방법으로 온톨로지가 사용된다. 온톨로지는 "도메인 지식을 명시적으로 개념화하는 명세서"로 정의될 수 있다[1]. 개념화는 용어와 용어들 간의 관계로 명시될 수 있고, 이렇게 명시된 온톨로지는 여러 DB상에서 질의를 할 때 서로 다

† 비 회 원 : 전북대학교 전산통계학과
khkim@chonbuk.ac.kr
†† 정 회 원 : 한국전자통신연구원 연구원
jhchoi@etri.re.kr
††† 비 회 원 : 전북대학교 전자정보공학부 교수
jdyang@chonbuk.ac.kr
†††† 비 회 원 : 한국전자통신연구원 지능형로봇연구단 연구원
bettle@etri.re.kr
논문접수 : 2004년 6월 19일
심사완료 : 2005년 4월 15일

른 DB내에서 나타나는 용어들 간의 불일치 문제를 해결함으로써, DB들 간의 연관성 추적, 통합, 재사용 그리고 공유를 가능하게 해준다[2-4].

특히, Gene Ontology(GO), MGED ontology, Open Biological Ontology 그리고 UMLS(Unified Medical Language System)등의 온톨로지는 생명정보학 분야에서 유전자의 기능을 예측하고 해석하는 많은 응용프로그램에서 사용된다[3,5-8]. 이러한 온톨로지는 생물학 정보에 대한 지식을 인간과 컴퓨터가 공유하고 사용할 수 있는 기반을 제공함으로써 생물학 지식의 이용을 극대화한다. GO에서는 생물학 지식을 3 분야(cellular component, biological process, molecular function)로 분류한다. 이 분야들은 각각 6933, 9018 그리고 1414개 용어로 구성되며, 이 용어들 사이는 일반화 계층 관계와 부분-전체 관계로 표현한다. UMLS는 미국 국립 의학도서관에서 개발한 의료분야의 정보 공유 및 교환을 위해 사용되고 있으며, 135개의 용어와 54개의 관계로 구성되어 있다. 그러나 GO는 관계를 기술하는 어휘가 일반화 계층과 부분-전체로 고정되어 있고, UMLS는 용어와 관계에 제약사항과 논리식 표현을 설정할 수 없기 때문에 복잡하게 구성되어 있는 생물학 용어들 사이의 관계를 정교하게 표현하기 어렵다.

한편, Protégé-2000[3,9]는 예술가 목록, 컴퓨터를 이용한 교육, 예술과 기술, 그리고 생명정보학 등의 많은 도메인을 지원하는 가장 널리 사용되는 온톨로지 개발 도구로 용어와 용어의 속성을 표현하기 위해 기호와 색을 이용한다. OilEd[4][10,11]은 DAML+OIL을 기반 언어로 쓰며 클래스들의 무결성과 암시적 포함 관계를 체크하기 위해 FaCT 추론기를 이용한다. Dag-Edit[12]는 GO를 편집하고 검색하기 위한 목적으로 개발된 편집기로 용어를 선택하면 그 용어의 모든 부모 용어를 모두 표시한다. 그러나 이상의 온톨로지 개발 도구들은 트리 구조의 그래픽 인터페이스를 사용하여 온톨로지를 구성하는 용어들을 계층구조로 표시하고 단순히 클래스, 인스턴스와 그들 간의 관계를 설정하는 기능만 지원한다. 따라서 복잡한 의미를 지닌 지식을 정교한 그래픽 표기를 이용하여 시각화하고 이를 쉽게 편집하는 목적에는 부합되지 않는 도구들로 볼 수 있다. ezOWL[13], OWLViz[14] 그리고 TGVizTab[15]은 Protégé-2000의 플러그인(Plugin)으로 개발된 온톨로지 시각화 도구로 위의 단점을 일부분 보완한다. ezOWL은 다이어그램 형식으로 온톨로지의 모든 관계를 하나의 화면에 모두 표시하고 정의된 그래픽 표기를 이용하여 온톨로지를 편집하는 기능을 지원한다. 그러나 전체 온톨로지 중에서 사용자가 관심 있는 부분만을 선별적으로 참조하고 다양한 방식으로 브라우징하는 기능은 지원하지 못한다. OWLViz는 온톨로지의 클래스 계층을, TGVizTab은 온톨로지의 클래스 사이의 관계를 화면에 표시하고 관련된 내용을 그래프로 표시한다. 하지만 단순히 클래스들 간의 관계만을 브라우징할 뿐이며 온톨로지 편집 기능은 제공하지 않는다.

한편 방대한 용어들로 구성되어 있는 온톨로지에서 용어간의 관계를 설정할 때 구축자는 온톨로지 전체 구조를 파악해야만 하는 부담을 갖게 된다. 그러나 지금까지 대부분 온톨로지 도구들은 이미 구축된 온톨로지의 내용을 구조적으로 파악하여 구축자에게 관련 후보 용어를 제시함으로써 이 부담을 경감시키는 기능을 지원하지 못한다.

본 논문에서는 OWL(Web Ontology Language)[16,17] 기반 그래픽 바이오 온톨로지 관리 시스템을 설계하고 구현하였다. 본 시스템에서 관리하는 온톨로지는 의미적 표현력이 뛰어난 OWL의 구조를 채택하기 때문에 복잡한 시멘틱을 가지는 생물학 지식을 정교하게 모델링할 수 있다. 또한, 본 시스템은 전문가가 시각화된 환경에서 쉽게 온톨로지를 관리할 수 있도록 새로운 온톨로지 그래픽 표기법과 반자동 구축 메커니즘을 지원한다. 즉, OWL의 강력한 의미적 표현력과 개념적으로 잘 정의한 그래픽 표기법들을 사용함으로써 정교한 바이오 온톨로지를 쉽게 관리할 수 있도록 지원한다.

본 논문의 구성은 다음과 같다. 2장에서는 생물학 지식을 표현하기 위한 온톨로지와 그래픽 표기법을 설명한다. 3장에서는 상속과 역상속 메커니즘을 이용하여 바이오 온톨로지를 구축하는 방법, 그리고 4장에서는 개발된 온톨로지 관리 시스템의 구현 내용을 각각 설명한다. 마지막으로 5장에서는 결론 및 향후 연구 방향에 대해 기술한다.

2. 생물학 지식 표현을 위한 온톨로지

본 논문에서 사용하는 OWL 기반 온톨로지는 다음과 같은 구조를 가지고 있다. 모든 온톨로지 내에 있는 용어는 클래스와 인스턴스로 정의되는데, 일반적인 의미의 클래스는 구체적인 의미의 클래스나 인스턴스를 하위 용어로 가진다. 인스턴스는 클래스의 구체적인 예, 또는 내용으로 실세계의 많은 용어는 클래스의 인스턴스로 정의된다. 온톨로지 용어 사이의 관계는 의미에 따라 일반화(super/subclass-of), 클래스화(owner/instance-of), 그리고 사용자 정의 관계(user-defined relationship)로 나뉜다. 따라서 하나의 용어는 수직적으로는 일반화, 클래스화 관계를 가지며, 수평적으로는 사용자 정의 관계를 가진다. 이때, 용어들 사이의 관계에 대한 제약사항은 속성(property)을 이용하여 설정한다.

그림 1은 생물학 지식을 표현한 온톨로지로 논문 전

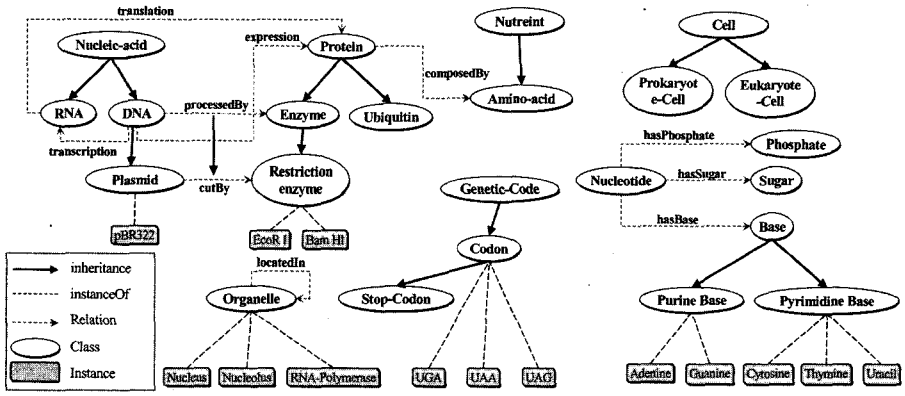

그림 1 생물학 지식 표현

반에 걸쳐 주요 예로 사용한다. 이 예에서 나타나는 온톨로지 용어들은 생물학 분야에서 자주 사용되는 용어로서, 타원으로 표시된 클래스와 둥근 사각형으로 표시된 인스턴스로 구분되며, 이들 사이의 관계는 용어사이의 간선으로 표시되었다.

클래스는 수직적으로 많은 하위 클래스와 일반화 관계를 가지며, 인스턴스들과는 클래스화 관계를 가진다[18]. 그림 1에서 'Nucleic-acid', 'Protein', 'DNA'는 클래스이며 'pBR322', 'EcoRI', 'BamHI'는 인스턴스이다. 또한 'DNA'는 'Nucleic-acid'의 하위 클래스로 일반화 관계가 설정되었으며, 'pBR322'는 'Plasmid'의 인스턴스로 클래스화 관계가 설정되었다. 용어 사이의 사용자 정의 관계는 도메인(domain)과 범위(range)를 이용하여 정의한다. 도메인은 온톨로지내의 용어가 되며, 범위는 온톨로지 용어 또는 일반적인 데이타 값이다. 전자는 범위에 클래스를, 후자는 범위에 정수형, 실수형 그리고 문자열 등과 같은 일반적인 데이타 값을 이용하여 정의한다. 그림 1에서 'DNA'와 'Enzyme' 사이에는 'processedBy'라는 사용자 정의 관계가 설정되었는데 'DNA'는 'processedBy'관계의 도메인으로, 'Enzyme'은 범위로 설정되었다.

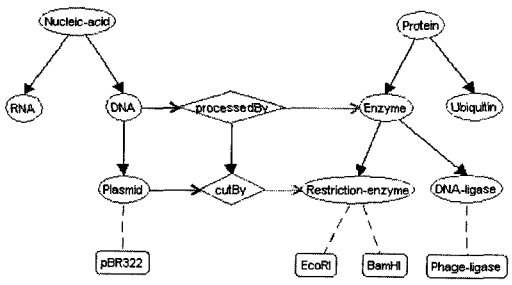

그림 2 그래픽 표기법을 이용한 온톨로지 표현

온톨로지를 효과적으로 시각화하여 표현하기 위해 본 논문에서 정의한 그래픽 표기법은 그림 2와 같다. 먼저, 온톨로지를 구성하는 기본 요소인 클래스, 인스턴스 그리고 사용자 정의 관계는 각각 ◯, ▭, ◇로 표현된다. 그림에서 보는바와 같이 'Nucleic-acid', 'DNA'는 클래스, 'pBR322', 'BamHI'는 인스턴스, 그리고 'processedBy', 'cutBy'는 관계이다. 'Nucleic-acid'와 'DNA'사이의 간선은 일반화 관계를 'Plasmid'와 'pBR322'사이의 간선은 클래스화 관계를 의미한다. 'processedBy'관계 좌우에 보이는 간선은 각각 도메인, 범위를 나타내며, 'processedBy'와 'cutBy'사이의 간선은 관계의 상속을 나타낸다. 즉, 두 용어 클래스 'DNA'와 'Enzyme'이 'processedBy' 관계를 가지기 때문에 이들의 하위 클래스인 'Plasmid'와 'Restriction-enzyme' 역시 'processedBy'의 일종인 'cutBy' 관계를 가지게 된다. 이 상속 메커니즘은 3장에서 자세히 설명한다.

온톨로지에서 용어들 사이의 관계에 속성(property)을 부여함으로써 이 관계를 의미적으로 모델링할 수 있다. 일반적으로 속성은 두 용어 사이에 설정된 관계에 대한 제약사항이 명시된다. 클래스에 설정할 수 있는 속성에는 합집합(Union), 교집합(Intersection), 여집합(Complement), 열거형(Enumeration)등의 논리식 표현(Logical Expression)이 있다. 예를 들어 그림 1과 같이 '세포(Cell)' 클래스는 '진핵세포(Eukaryote-Cell)'와 '원핵세포(Prokaryote-Cell)'를 하위 클래스로 가지기 때문에 '세포'와 이 두 클래스는 일반화 관계를 가진다. 그런데, 이 관계에 그림 3과 같이 합집합이라는 속성을 부여함으로써 '세포' 클래스는 의미적으로 '진핵세포'와 '원핵세포'외에 어떤 클래스도 하위 클래스로 가질 수 없다는 제약 조건을 표현하게 된다. 이 속성은 온톨로지 구

축 과정에서 전문가가 '세포'에 다른 하위 클래스를 삽입할 수 없도록 제한함으로써 온톨로지의 의미적 일관성을 유지하게 된다. 또한 '종료 코돈(Stop-Codon)' 클래스와 실제 '종료 코돈' 인스턴스들 사이의 속성은 열거형으로 설정할 수 있다. 따라서 모두 64개의 '코돈' 중에서 명시한 'UGA', 'UAA' 그리고 'UAG'만이 '종료 코돈'에 속한다는 지식을 표현할 수 있다. 이와 같이 속성을 이용하면 계층적인 의미의 생물학적 지식을 구체적인 클래스 계층으로 정교하게 구성할 수 있다. 합집합, 교집합, 여집합 그리고 열거형에 대한 속성은 각각 ⓤ, ⓝ, ⓒ 그리고 ⓘ로 표현된다.

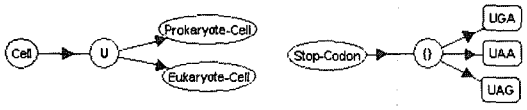

그림 3 클래스에 속성 설정

용어 사이의 관계는 전이성(Transitive), 대칭성(Symmetric), 함수성(Functional), 역(Inverse) 그리고 역함수성(InverseFunctional) 등의 속성과 모든값(∀), 임의값(∃), 관계 차수(Cardinality)와 같은 속성값을 제한하는 제약사항으로 설정할 수 있다. 관계 속성 중 전이적 속성을 예로 들어 설명하면 다음과 같다. 'locatedIn' 관계가 도메인과 범위에 '세포소기관(Organelle)'으로 정의되고 전이적 속성이 설정되었을 때, "'핵(Nucleus)'속에 '인(Nucleolus)'이 있고, '인' 속에 'RNA-Polymerase'가 있다"라고 정의되어 있을 경우 'locatedIn'에 전이적 속성이 설정되어 있으므로 "'핵' 속에 'RNA-Polymerase'가 있다"라고 추론을 통해 추가적인 지식을 얻을 수 있다. 그림 4의 'locatedIn' 관계는 도메인과 범위로 'Organelle', 속성으로는 전이적 속성을 설정한 예이다.

그림 4 관계에 속성 설정

그림 5는 'DNA'와 'RNA'사이에 'transcription' 관계, 'RNA'와 'Protein'사이에 'translation'관계, 그리고 'DNA'와 'Protein'사이에 'expression'관계가 설정된 것을 그래픽 표기법으로 표현하였다. 그래픽 표기법을 이용하면 "DNA는 RNA로 전사(transcription)되고, RNA는 단백질(Protein)로 해석(translation)된다. 즉 DNA는 단백질로 발현(expression)된다"라는 지식을 쉽게 표현할 수 있다. 또한 그림을 보면 'expression' 관계가 'transcription'과 'translation'의 복합관계로 구성되어 있음을 알 수 있다.

그림 5 복합 관계 표현

제약사항을 관계에 설정하여 정의하면 관계를 보다 자세하게 설정할 수 있다. "유비키틴(Ubiquitin)은 76개의 아미노산(Amino-acid)으로 이루어져있다"라는 지식은 용어와 관계만으로는 표현이 불가능하다. 'composedBy'라는 관계를 '단백질' 도메인과 '아미노산' 범위를 이용하여 정의할 수 있지만 "76개의 아미노산"이라고 한정적으로 정의할 수 없다. 이는 제약사항을 이용하여 다음과 같이 정의할 수 있다. 관계 'composedBy'에 제약사항을 적용하여 'composedBy'의 범위 값으로 '아미노산'만 올 수 있도록 하고 '아미노산'의 개수를 관계 차수를 이용하여 76개로 제약사항을 설정하여 표현할 수 있다. 정리하면 "유비키틴은 단백질의 일종으로 아미노산으로 이루어져있는데 76개로 이루어져 있다"라는 지식을 나타낸다. 그림 6은 위의 지식을 그래픽 표기법을 이용해서 표현한 것이다. 제약사항에 속하는 관계 'composedBy'는 onProperty 간선에 그리고 제약사항 설정 사항들은 allValuesFrom과 cardinality 간선에 표시된다.

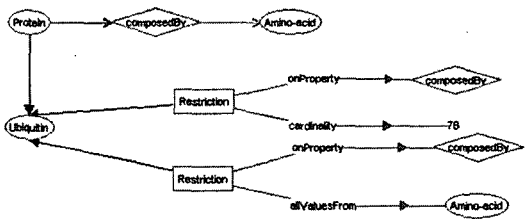

그림 6 관계에 제약사항 설정

위와 같이 본 논문에서 정의한 그래픽 표기법을 이용하면 생물학 분야의 복잡한 지식도 몇 개의 그래픽 표기법으로 간략하게 표현되기 때문에 사용자가 온톨로지의 내용을 직관적으로 편집하고 참조할 수 있다.

3. 바이오 온톨로지 구축

바이오 온톨로지에 표현되는 방대한 용어들 사이의 복잡한 관계들은 도메인 전문가가 이들의 의미적 구조를 전체적으로 파악할 수 없게 하여, 일관성 있는 온톨로지 구축과 관리를 어렵게 한다. 이 장에서는 OWL 기

반 온톨로지를 이용하여 생물학 용어들 사이의 복잡한 관계를 의미적 손실 없이 쉽게 구축할 수 있는 방법을 설명한다.

3.1 클래스 용어 관계 구축

온톨로지에서 클래스 용어들은 그 클래스에 속한 많은 인스턴스 용어들을 대표하게 된다. 따라서 온톨로지의 방대한 인스턴스들을 체계적으로 구축하기 위해서는 이 인스턴스들이 속해 있는 클래스들의 관계나 속성들을 일관성 있게 설정할 필요가 있다. 그러나, 온톨로지는 일반적으로 여러 전문가들에 의해 구축되기 때문에 일관성 있게 온톨로지를 관리하기 위해 한 구축 전문가가 모든 클래스들의 관계를 파악하고 있기는 매우 어렵다. 이 단점을 보완하기 위해 본 논문에서는 [19]에서 제안한 상속에 의한 구축 방법을 OWL 기반 온톨로지에 적합한 형태로 적용하였다.

예를 들어, 그림 7에서 다른 전문가에 의해 'Enzyme'에 관련된 온톨로지 용어 관계가 구축되어 있으며, 현재 'DNA'에 관련된 온톨로지 용어 관계를 구축하고 있다고 가정하자. 이때, 전문가는 'Enzyme'의 용어 관계를 정확히 파악해야 'DNA'의 하위 클래스인 'Plasmid'와의 관계를 명확히 구축할 수 있다. 그러나 여러 전문가들에 의해 구축된 온톨로지 용어 관계를 현재 구축하고 있는 전문가가 정확히 파악하기는 매우 어렵다. 상속 메커니즘은 이들의 관계 정보를 반자동으로 파악할 수 있는 매우 유용한 방법으로 이용될 수 있다. 즉, 'DNA'와 'Enzyme' 사이에 'processedBy'라는 사용자 정의 관계가 정의되어 있을 때, 이 관계는 상속의 성질을 이용하여 하위 클래스들의 관계로 자동 상속된다.

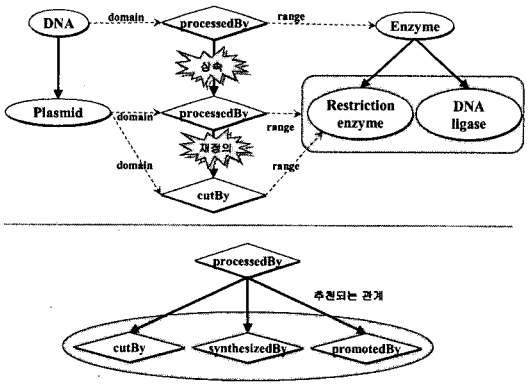

그림 7 관계 상속

'DNA'의 하위 클래스로 'Plasmid'가 추가되었을 경우 관계 상속의 과정은 다음과 같다. 먼저 'processedBy' 관계는 도메인은 'DNA', 범위는 'Enzyme'으로 설정되어 있다. 이 'processedBy' 관계는 도메인과 범위의 하위 클래스들의 관계로 자동으로 상속된다. 따라서 'Plasmid'는 'Restriction-enzyme'과 'DNA-ligase' 사이에 'processedBy'라는 잠정적인 관계를 자동으로 가지게 된다. 이때 전문가가 'Restriction-enzyme'과 'DNA-ligase'에 대해 'processedBy'보다 구체적인 관계를 명시하고자 할 경우, 상속에 의한 메커니즘은 'processedBy' 관계의 계층도를 파악하여 'processedBy'의 하위 관계인 'cutBy', 'synthesizedBy', 그리고 'promotedBy' 관계들을 구축 전문가에게 추천하여 제시한다. 추천된 관계들 중 'cutBy' 관계를 선택하면 이 관계는 'processedBy' 관계의 하위 관계로 도메인을 'Plasmid', 범위를 'Restriction-enzyme'으로 삽입된다. 이 관계 재정의는 'Plasmid'와 'Restriction-enzyme'에 소속된 많은 인스턴스들 사이에 존재하는 잠정적인 관계를 보다 구체적인 관계로 설정할 수 있도록 한다. 따라서 상속 메커니즘에 의한 구축 방법은 클래스들 사이의 관계를 일관성 있게 명시할 수 있도록 하며 이 클래스에 속한 하위 클래스나 방대한 인스턴스들의 잠정적 관계를 자동으로 명시할 수 있도록 한다.

3.2 인스턴스 용어 관계 구축

온톨로지에서 인스턴스들은 가장 구체적인 의미의 용어들로 클래스보다 빈번히 변화되며, 특정 응용 도메인에서 많이 이용된다는 특징을 가진다. 따라서 온톨로지를 구축하고 유지할 때, 인스턴스들의 관계를 특히 정확히 명시해야 한다. 그러나 일반적으로 바이오 온톨로지에서 하나의 클래스는 매우 많은 인스턴스들을 포함하고 있기 때문에 이들을 구축 전문가가 일일이 명시하기는 매우 어렵다. 본 논문에서 제안하는 온톨로지 구축 방법은 클래스들에 설정된 관계 정보를 통해 이 클래스에 속한 인스턴스의 관계를 구조적으로 명시할 수 있다는 특징을 가지고 있다.

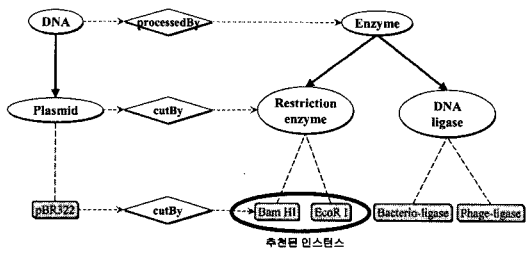

그림 8 인스턴스 관계 설정

예를 들어, 그림 8은 'pBR322'와 'BamHI'사이에 'cutBy' 관계가 설정되는 과정을 설명하고 있다. 'pBR322'는 소유 클래스(owner)인 'Plasmid'와 부모

클래스인 'DNA'에 설정된 각각 'cutBy'와 'processedBy' 관계를 이용해서 인스턴스간의 관계를 설정할 수 있다. 즉, 'processedBy' 관계일 경우 대상되는 후보 인스턴스는 'processedBy'의 범위인 'Enzyme' 클래스와 'Enzyme' 클래스의 하위 클래스들의 모든 인스턴스 'BamHI', 'EcoRI', 'Bacterio-ligase' 그리고 'Phage-ligase'가 된다. 반면, 'cutBy' 관계일 경우 대상되는 후보 인스턴스는 'cutBy'의 범위인 'Restriction-enzyme' 클래스의 모든 인스턴스 'BamHI'와 'EcoRI'가 된다. 이러한 방식으로 시스템은 해당 관계에 설정될 수 있는 후보 인스턴스들을 구조적으로 파악하여 구축 전문가에게 제시하게 된다. 이들 중에서 전문가가 'cutBy' 관계에 의해 추천된 후보 인스턴스들 중 'BamHI'를 선택하면 'pBR322'와 'cutBy' 관계가 명시적으로 설정된다. 즉, 소유 클래스의 관계뿐만 아니라 소유 클래스의 부모 클래스의 관계들을 이용하여 관계를 설정할 수 있다. 이 구축 방법은 온톨로지 전문가가 방대한 용어들 사이의 관계들을 일관성 있게 유지시킬 수 있도록 지원한다.

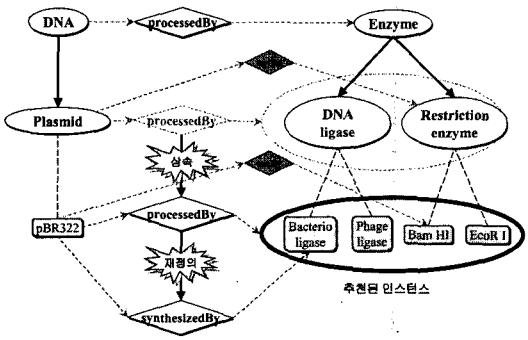

그림 9 인스턴스 관계 재정의

그림 9에서와 같이 'pBR322'와 'Bacterio-ligase'사이에 'synthesizedBy' 관계를 설정하고자 할 경우 보다 복잡한 상속 메커니즘이 요구된다. 그 이유는 두 클래스 'Plasmid'와 'DNA-ligase' 사이에 단지 잠정적인 'processedBy' 관계만이 설정되어 있고 구체적인 'synthesizedBy' 관계가 설정되어 있지 않기 때문이다. 따라서, 'pBR322'의 소유 클래스 'Plasmid'와 잠정적 관계 'processedBy'를 가지는 'DNA-ligase'와 'Restriction-enzyme'에 소속된 인스턴스 'Bacterio-ligase', 'Phage-ligase', 'BamHI' 그리고 'EcoRI'들이 모두 추천된다. 구축 전문가는 이 추천된 인스턴스들 중에서 'Bacterio-ligase'를 'synthesizedBy' 관계로 재정의할 수 있다.

그런데, 잠정적인 'processedBy' 관계에 의해 추천되는 'DNA-ligase'와 'Restriction-enzyme'의 인스턴스가 매우 많을 경우, 전문가가 이들 중 적절한 인스턴스를 선택하기 어려운 문제점이 발생하게 된다. 이 문제점은 다른 구축 전문가가 'Plasmid'에 인스턴스를 추가할 때 마다 매번 발생하게 된다. 따라서, 이 경우 구조적으로 인스턴스들 사이에 설정된 구체적인 관계를 이용하여 클래스들 사이의 관계에 대한 종류를 추천할 수 있는 역상속 메커니즘이 필요하다. 즉, 구축 전문가가 잠정적으로 설정된 클래스들의 관계를 명시적인 관계로 설정하기 위해 인스턴스들 사이에 설정된 구체적인 관계를 이용할 수 있다.

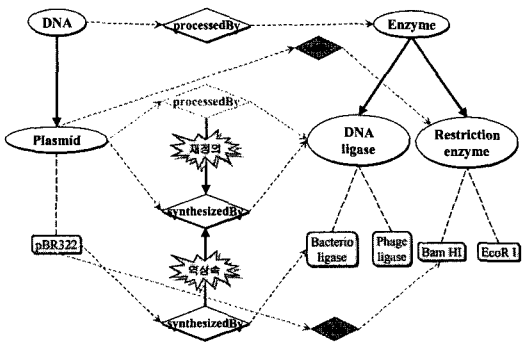

그림 10 역상속 메커니즘

예를 들어, 그림 10과 같이 두 클래스 'Plasmid'와 'DNA-ligase' 사이에는 'processedBy'라는 잠정적 관계가 설정되어 있고 두 인스턴스 'pBR322'와 'Bacterio-ligase' 사이의 관계는 전문가에 의해 'synthesizedBy'로 명시되었다고 가정하자. 이때, 클래스 'Plasmid'와 'DNA-ligase'는 각각의 클래스에 속한 모든 인스턴스들의 속성을 대표한다고 할 수 있기 때문에 두 인스턴스 'pBR322'와 'Bacterio-ligase' 사이에 설정된 관계에 대한 속성 역시 이들 각각의 클래스에 포함되어 있어야 한다. 따라서 인스턴스 'pBR322'와 'Bacterio-ligase' 사이의 관계 'synthesizedBy'를 설정할 때, 이들을 포함하고 있는 두 클래스 'Plasmid'와 'DNA-ligase' 사이에 잠정적 관계 'processedBy'를 명시적 관계 'synthesizedBy'로 재설정할 수 있다는 구조적 정보를 구축 전문가에게 제시할 수 있다. 즉, 인스턴스들 사이의 관계가 이들이 속한 클래스 정보에 역상속되었다고 할 수 있다. 이 역상속 메커니즘에 의해 'Plasmid'와 'DNA-ligase' 사이에 'synthesizedBy' 관계가 설정되었기 때문에 'Plasmid'에 새로 추가되는 많은 인스턴스들에 대한 'synthesizedBy' 관계를 단순한 상속 메커니즘을 통해 설정할 수 있게 된다. 따라서 역상속 메커니즘은 구축 과정에 있는 전문가에게 잠정적인 클래스들 사이의 관계를 명시적 관계로 일관성 있게 설정할

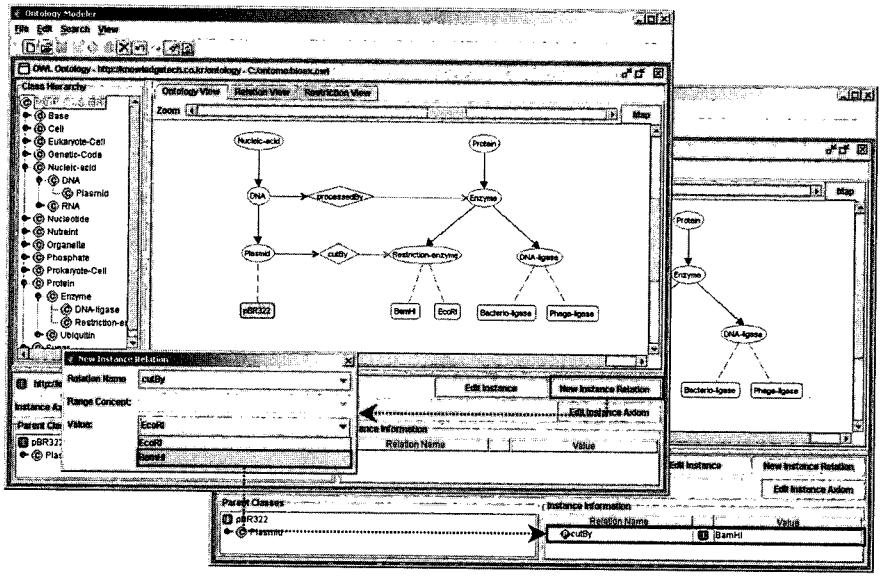

그림 11 인스턴스 관계 설정

수 있도록 함으로써 이들에 포함된 많은 인스턴스들의 관계를 쉽게 설정할 수 있도록 지원한다.

4. 구 현

이 장에서는 앞에서 예를 든 생물학 지식을 표현한 OWL 기반의 온톨로지를 그래픽 환경에서 보다 쉽게 관리할 수 있도록 지원하는 온톨로지 관리 시스템의 구현 내용에 대해 기술한다. 이 시스템은 온톨로지 구축 기능, 브라우징 기능 그리고 구축된 온톨로지를 다양한 온톨로지 언어로 변환하는 기능 등을 포함한다. 이 기능들은 Java 환경에서 구현되었으며, 온톨로지를 시각화하기 위해 TouchGraph[20]을 사용하였다. 구축된 온톨로지 정보는 객체관계형 데이타베이스나 구조화된 포맷을 가지는 파일로 각각 저장된다.

4.1 온톨로지 구축 기능

이 절에서는 시스템의 온톨로지 구축 기능을 3장에서 설명한 인스턴스 관계 구축에서 제시한 시나리오를 기반으로 기술한다. 클래스 용어에 대해서도 역시 같은 방법을 적용하여 구축할 수 있다. 이 시나리오는 시스템에 의해 잠정적으로 설정되는 인스턴스 관계 구축 방법, 잠정적인 관계를 보다 구체적인 의미의 관계로 재정의 하는 과정, 역상속에 의한 잠정적인 클래스 관계를 재설정 그리고 관계에 제약 조건을 설정하는 과정으로 구성되어 있다.

그림 11에서 'pBR322'와 'BamHI' 사이의 관계를 상속 메커니즘을 통해 설정하는 과정을 나타내고 있다. 즉 'pBR322'의 상위 클래스 'Plasmid', 'DNA' 그리고 'Nucleic-acid'가 다른 클래스와 가지는 관계 'cutBy'와 'processedBy'를 Relation Name에 제시하고 있다. 이때, 'cutBy'를 선택하면, 'Plasmid'와 'cutBy' 관계를 가지고 있는 'Restriction-enzyme'의 인스턴스들 'BamHI'와 'EcoRI'가 'pBR322'의 'cutBy' 범위에 대한 후보 인스턴스로 추천된다. 같은 방법으로 'processedBy'를 선택했을 경우 'Restriction-enzyme'과 'DNA-ligase'의 인스턴스들 'BamHI', 'EcoRI', 'Bacterio-ligase' 그리고 'Phage-ligase'가 추천된다. 전문가는 'cutBy' 범위에 있는 추천된 후보 인스턴스들 중에서 'BamHI'를 선택하면, 두 인스턴스의 관계가 명시적으로 설정된다. 이 시스템에 의해 제공되는 상속에 의한 관계 설정 메커니즘은 방대한 인스턴스들 사이의 관계를 반자동으로 구축할 수 있게 할 뿐만 아니라 여러 전문가에 의해 구축되는 온톨로지에 대한 의미적 일관성을 유지시킬 수 있는 특징을 가진다.

그림 12에서 'DNA'와 'Enzyme' 사이에 'processedBy' 관계가 설정되어 있기 때문에 'pBR322'와 'Bacterio-ligase' 역시 잠정적으로 설정된 'processedBy' 관계를 가지고 있다. 이 시스템은 이 포괄적이고 잠정적인 관계를 구체적이고 명시적인 관계 'synthesizedBy'로 재정의할 수 있다. 즉, 'pBR322'가 소속된 클래스 'DNA'와 'processedBy' 관계를 가지는 'Enzyme'의 모든 인스턴스 'BamHI', 'EcoRI', 'Phage-ligase' 그리고 'Bacterio-ligase'를 구조적으로 파악하

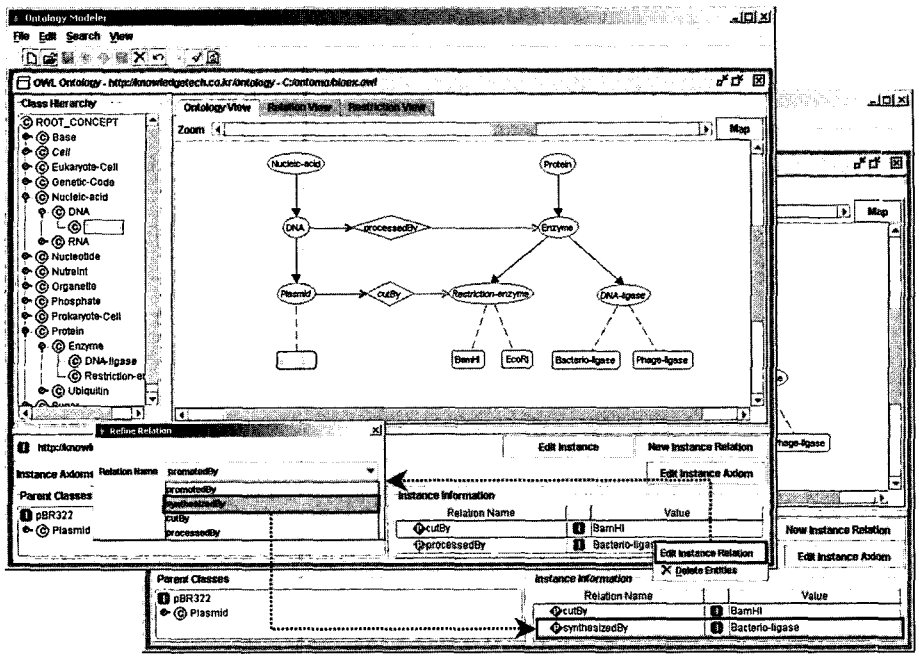

그림 12 인스턴스 관계 재정의

여 전문가에게 제시된다. 이들 중 전문가는 'pBR322'와 실제 'synthesizedBy' 관계를 가지는 'Bacterio-ligase'를 선택하면, 시스템은 Edit Instance Relation 메뉴를 통해 이 두 인스턴스가 가질 수 있는 구체적인 후보 관계를 추천한다. 이 후보 관계들은 두 인스턴스들이 각각 소속된 'DNA'와 'Enzyme' 사이의 관계 'processedBy'에 의해 파악될 수 있다. 즉, 온톨로지에 포함된 'processedBy'의 관계 계층을 구조적으로 파악하여 그 하위 관계인 'synthesizedBy', 'cutBy' 그리고 'promotedBy'등을 추천한다. 이 추천된 관계들 중에서 구축 전문가가 'synthesizedBy'를 선택함으로써 두 인스턴스 사이의 관계를 구체화할 수 있다.

그림 13은 두 인스턴스 'pBR322'와 'Bacterio-ligase' 사이에 설정된 'synthesizedBy' 관계를 통해 이들의 부모 클래스들 사이의 관계를 구체화하는 역상속 메커니즘을 설명하고 있다. 즉, 시스템은 'pBR322'와 'Bacterio-ligase' 사이에 'synthesizedBy' 관계가 설정되었기 때문에 이들의 소속 클래스 'Plasmid'와 'DNA-ligase' 사이에도 'synthesizedBy' 관계가 설정될 수 있음을 전문가에게 제안한다. 전문가가 이 제안된 관계가 적절할 경우 'Plasmid'와 'DNA-ligase'사이에 명시적인 관계 'synthesizedBy'를 설정한다. 이 시스템에 의해 구현된 역상속 메커니즘은 인스턴스들 사이의 관계를 통해 이들이 소속된 클래스들 사이의 관계를 구체적으로 명시할 수 있다. 따라서, 향후 다른 구축 전문가에 의해 이들 클래스에 추가되는 방대한 인스턴스들 사이의 관계를 일관되게 명시할 수 있다.

그림 14는 Restriction View에서 'Nucleotide' 클래스에 대한 제약사항을 설정하는 과정을 설명하고 있다. 즉, 이 제약사항은 onProperty 속성으로 'hasSugar' 관계가 그리고 MinCardinality 값으로 1을 갖도록 설정될 수 있다. 즉 'Nucleotide'의 인스턴스는 1개의 'Sugar' 인스턴스와 관계를 설정할 수 있다. 각각의 클래스에 제약사항을 설정할 수 있게 함으로써 구축 과정에서 온톨로지의 일관성을 구체적으로 유지시킬 수 있도록 한다. 본 시스템은 위와 같이 복잡한 지식을 시각화된 구축 메커니즘을 통해 쉽게 온톨로지로 표현할 수 있다는 장점을 가지고 있다.

4.2 브라우징 기능

온톨로지 관리 시스템에서 브라우징 기능은 여러 전문가들에 의해 구축된 온톨로지를 쉽게 파악할 수 있도록 지원한다. 즉, 트리 형태의 온톨로지 클래스 계층도를 이용하여 상위/하위 관계를 항해하며 클래스 정보를 브라우징할 수 있다. 이때, 각각의 클래스를 선택할 때마다 화면 아래에 선택한 클래스와 관련된 정보가 실시간으로 변경되어 온톨로지 구축자에게 제시된다. 또한, 온톨로지의 계층에 포함된 용어들 사이의 관계에 대한 자세한 정보들은 정의된 그래픽 표기법을 통해 시각화된 환경으로 제공된다.

그림 15는 일반적인 의미의 클래스로부터 구체적인

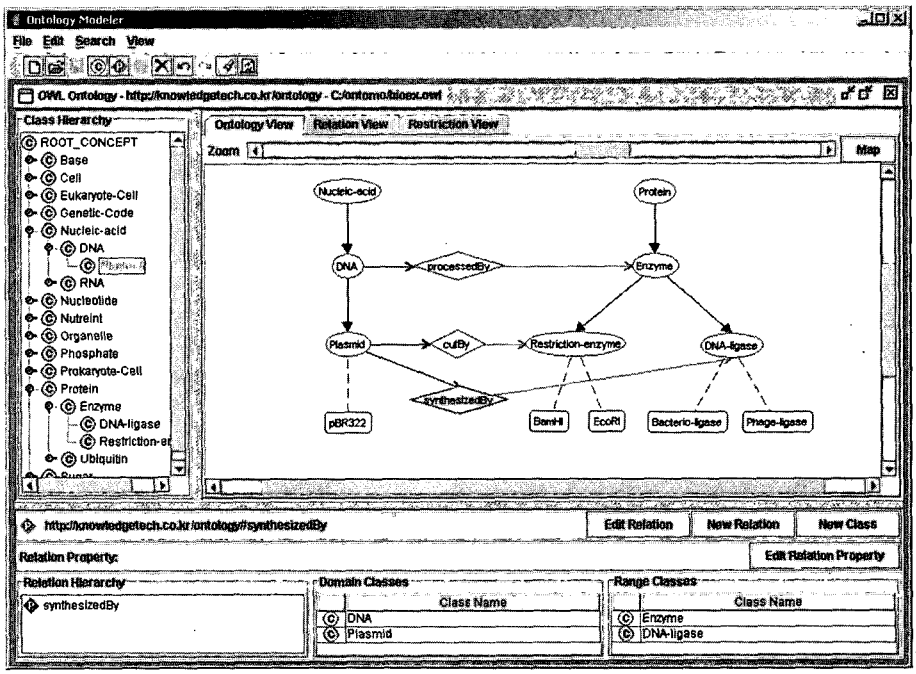

그림 13 관계 역상속 후 결과

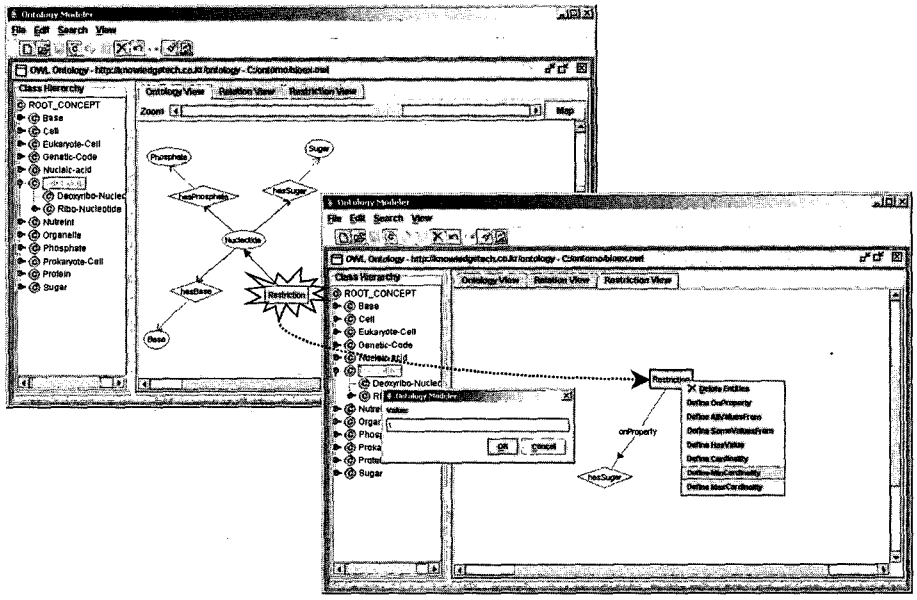

그림 14 제약사항 설정

의미의 클래스 또는 인스턴스로 점진적으로 브라우징하는 과정을 나타낸다. 이 기능은 하나의 최상위 클래스로부터 파생되는 많은 하위 클래스들 중에서 사용자가 의도한 방향에 있는 클래스들만을 순차적으로 브라우징할 수 있게 한다. 즉, 일반적인 의미의 'Base' 클래스로부터 하위 클래스인 'Phyrimidine-Base' 그리고 'DNA-Phyrimidine-Base'로 점진적으로 브라우징할 수 있다.

또한, 본 시스템은 브라우징된 온톨로지에 대한 3 가지 시각화 뷰를 제공한다. 즉, 클래스 중심의 Ontology View, 용어 사이의 관계 중심의 Relation View 그리고

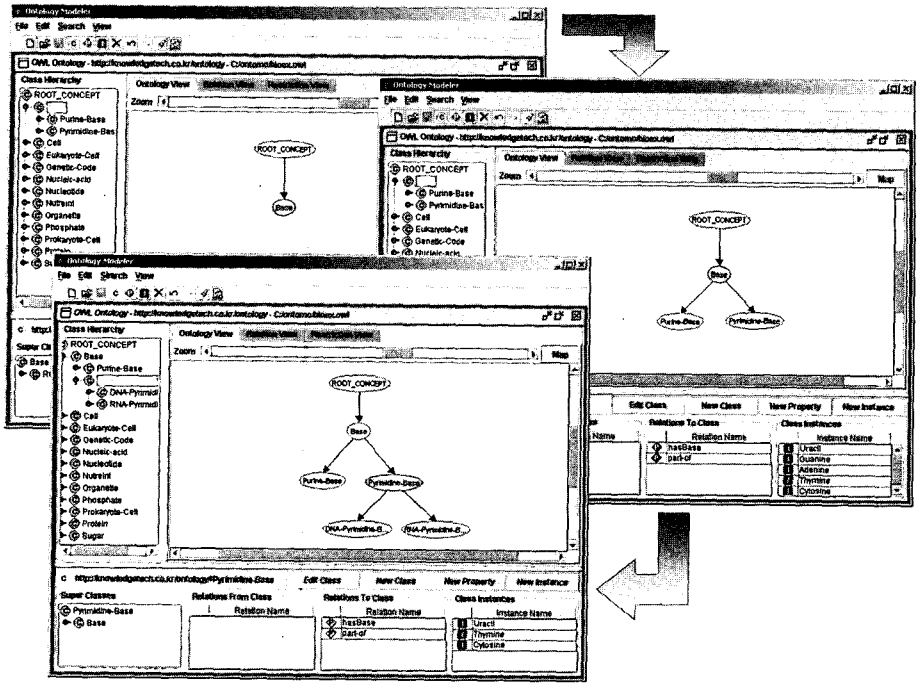

그림 15 점진적 브라우징

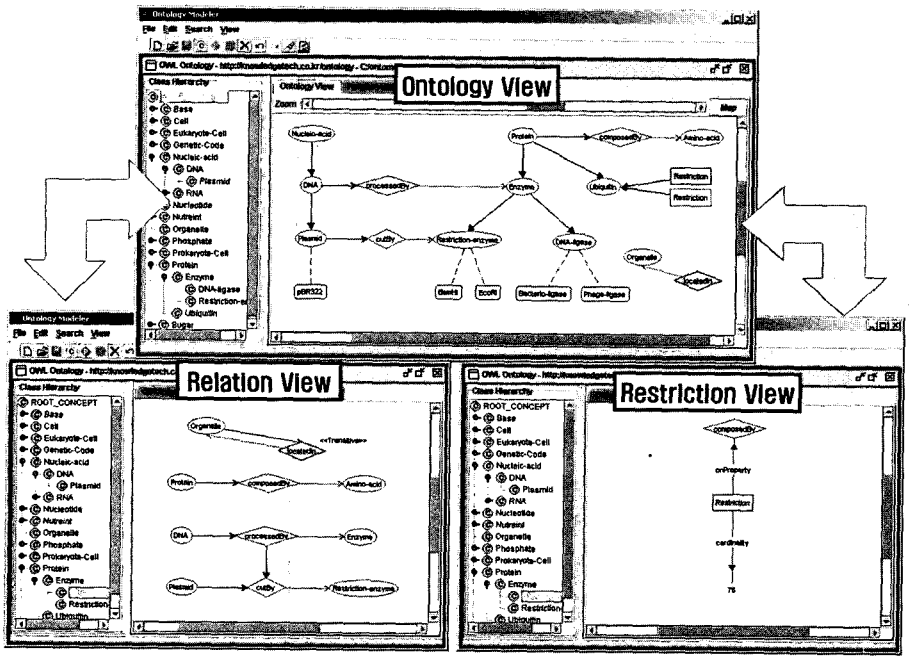

그림 16 클래스/관계/제약사항 중심 브라우징

클래스 제약을 관리할 수 있는 Restriction View를 제공한다. 예를 들어, 그림 16에서 사용자가 자신이 참고하고자하는 클래스 'Nucleic-acid', 'Protein', 'Amino-acid' 그리고 'Organelle'등을 선택하여 브라우징하면, 이들이 속한 온톨로지 계층 그리고 이 계층에 포함된 용어들 사이의 관계가 Ontology View 화면에 시각화된

다. 이 Ontology View에서 표시된 사용자 정의 관계 'locatedIn', 'composedBy', 'processedBy' 그리고 'cutBy'들에 대한 정보와 이들의 속성은 Relation View를 통해 시각화할 수 있다. 특히, 이 Relation View에서 'cutBy'는 'processedBy'의 하위 관계임을 그리고 'locatedIn'은 전의적 속성이 설정되었음을 나타내고 있다. 또한, Ontology View에서 정의된 클래스의 제약사항을 자세히 편집하고 참조할 수 있도록 Restriction View 역시 제공한다. 이 뷰를 통해 현재 'Ubiquitin'에 cardinality가 76으로 설정되었음을 참조할 수 있다. 온톨로지가 GO와 UMLS와 같이 많은 용어와 관계로 복잡하게 구성되어 있을 경우, 사용자는 점진적 브라우징이 가능한 이 3가지 시각화 뷰들을 통해 자신이 관심이 있는 용어와 관계들을 쉽게 파악하여 구축할 수 있게 한다. 이때, 확대/축소가 가능한 온톨로지 맵 기능은 온톨로지에 대한 전체적인 구조를 간단하게 시각화한다.

4.3 온톨로지 언어 생성 기능

본 논문에서 구현된 시스템은 JENA[21]을 채용함으로써 전문가에 의해 구축된 온톨로지를 RDFS, DAML+OIL, OWL 등 여러 표준화된 형식으로 변환할 수 있다. 이 변환은 다른 시스템에 의해 여러 형식으로 구축된 온톨로지들을 본 시스템으로 통합할 수 있도록 지원한다. 특히, 구축된 온톨로지는 OWL로 완벽하게 변환이 가능한데, OWL은 W3C에서 권고안으로 발표된 차세대 웹 온톨로지 언어로 기존의 RDF/RDFS, DAML+OIL에 비해 클래스와 관계에 대하여 기술할 수 있는 더 많은 어휘를 제공한다. 예를 들면, 클래스간의 관계, 관계 차수, 동치성, 풍부한 속성 타입, 열거형 클래스등을 기술할 수 있다. 따라서 기존의 온톨로지 언어보다 의미적 표현력이 상대적으로 뛰어나다.

이전 장에서 설명한 "유비키틴은 76개의 아미노산으로 이루어져있다"라는 지식은 그림 17과 같이 OWL 파일로 생성된다. 생성된 OWL 파일 내용은 다음과 같다. 먼저 'composedBy'라는 관계를 '단백질' 도메인과 '아미노산' 범위를 이용하여 정의한다. 다음에는 '유비키틴' 클래스를 '단백질' 클래스의 하위 클래스로 정의함과 동시에 제약사항을 이용하여 이전에 정의된 'composedBy' 관계를 보다 자세히 재정의 한다. allValuesFrom을 이용하여 'composedBy'의 범위값에 '아미노산' 인스턴스만 올 수 있도록 하고 cardinality를 이용하여 'composedBy' 관계에 76개의 인스턴스로 설정할 수 있게 제약을 가한다.

그림 18은 "발현은 전사와 해석을 통해 이루어진다"라고 정의된 온톨로지 지식이 OWL로 생성된 예이다. 'DNA'는 'RNA'로 '전사'되고 'RNA'는 '단백질'로 '해석'된다. 즉 'DNA'는 '단백질'로 '발현'된다. 생성된 OWL에 '발현' 관계는 '전사'와 '해석'의 복합 관계로 구성되어 있음을 알 수 있다. 이와 같이 본 시스템을 이용하여 구축한 지식이 다양한 표준 온톨로지 언어로 변환됨으로써 다양한 온톨로지 시스템에서 쉽게 공유되어 사용될 수 있으며 시멘틱 웹 응용에 바로 적용될 수 있다.

```
<owl:ObjectProperty rdf:ID="composedBy">
  <rdfs:domain rdf:resource="#Protein"/>
  <rdfs:range rdf:resource="#Amino-acid"/>
</owl:ObjectProperty>
<owl:Class rdf:ID="Ubiquitin">
  <rdfs:subClassOf rdf:resource="#Protein"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#composedBy"/>
      <owl:allValuesFrom rdf:resource="#Amino-acid"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#composedBy"/>
      <owl:cardinality>76</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

그림 17 Cardinality를 이용한 제약

```
<owl:ObjectProperty rdf:ID="transcription">
  <rdfs:domain rdf:resource="#DNA"/>
  <rdfs:range rdf:resource="#RNA"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="translation">
  <rdfs:domain rdf:resource="#RNA"/>
  <rdfs:range rdf:resource="#Protein"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="expression">
  <rdfs:domain rdf:resource="#DNA"/>
  <rdfs:range rdf:resource="#Protein"/>
</owl:ObjectProperty>
```

그림 18 복합 관계 표현

5. 결 론

본 논문에서는 시각화된 환경에서 정교한 바이오 온톨로지를 표현하고 구축하기에 적합한 그래픽 온톨로지 관리 시스템을 제안하였다. 상속 메커니즘에 의한 구축 방법에 의해 구조적으로 파악된 정보는 온톨로지 구축 시 반자동 방식을 지원하며, 온톨로지의 의미적 일관성을 유지시켜준다. 역상속 메커니즘은 구축과정에 있는 전문가에게 잠정적인 클래스들 사이의 관계를 명시적으로 일관성 있게 설정할 수 있도록 한다. 또한 많은 용어와 관계들로 복잡하게 구성되는 생물학 지식을 정의된 그래픽 표기법을 이용하여 간단한 조작만으로 쉽게 바이오 온톨로지를 편집하고 참조할 수 있는 온톨로지 관리 시스템을 구현하였다.

향후 연구로는 구현된 온톨로지 관리 시스템에 질의 시스템을 추가하여, 브라우징뿐만 아니라 질의를 이용하여 도메인 지식을 추론하고 검색함으로써 보다 쉽게 지식을 참조할 수 있는 연구가 필요하다. 또한 많은 노드와 간선으로 이루어진 온톨로지를 효율적으로 시각화할 수 있는 연구가 필요하다.

참 고 문 헌

[1] Gruber, T., "A Translation Approach to Portable Ontology Specifications," in Knowledge Acquisition Journal, Vol. 5, pp. 199-220, 1993.

[2] Lambrix, P., Habbouche, M. and Perez, M., "Evaluation of ontology development tools for bioinformatics," Bioinformatics, Vol. 19, pp. 1564-1571, 2003.

[3] Rosse, C. and Megino, J.L.V., "A reference ontology for bioinformatics : the Foundational Model of Anatomy," Journal of Biomedical Informatics, Inpress, 2003.

[4] Stevens, R., Goble, C., Horrocks, I. and Bechhofer, S., "Building a Bioinformatics Ontology Using OIL," IEEE Transactions on Information Technology in Biomedicine, Vol. 6, pp. 135-141, 2002.

[5] Jensen, L.J., Gupta, R., Starfeldt, H.-H. and Brunak, S., "Prediction of Human Protein Function According to Gene Ontology Categories," Bioinformatics, Vol. 19, pp. 653-642, 2003.

[6] Karp, P.D., "An Ontology for Biological function based on molecular interactions," Bioinformatics, Vol. 16, pp. 269-285, 2000.

[7] Westbrook, J.D. and Bourne, P.E., "STAR/mmCIF: Anontology for macromolecular structure," Bioinformatics Ontology, Vol. 16, pp. 159-168, 1999.

[8] Kumar A., Smith B., "The Unified Medical Language System and the Gene Ontology: Some Critical Reflections," Proc KI 2003, pp. 135-148, 2003.

[9] Noy, N. F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R. W. and Musen, M. A., "Creating Semantic Web Contents with Protege-2000," IEEE Intelligent Systems 16(2), pp. 60-71, 2001.

[10] Bechhofer, S., Horrocks, I., Goble, C. and Stevens, R., "OilEd: a Reason-able Ontology Editor for the Semantic Web," Proceedings of KI2001, LNAI Vol. 2174, pp. 396-408, 2001.

[11] Horrocks, I., "Reasoning with Expressive Description Logics:Theory and Practice," Proceedings of CADE-02, Springer-Velag Lecture Notes in Artificial Intelligence, LNAI 2393, pp. 1-15, 2002.

[12] DagEdit, <http://geneontology.sourceforge.net/>

[13] ezOWL <http://iweb.etri.re.kr/ezowl/index.html>

[14] OWLViz <http://www.co-ode.org/downloads/owlviz/>

[15] TGVizTab <http://www.ecs.soton.ac.uk/~ha/TGVizTab/TGVizTab.htm>

[16] Dean, M. and Schreiber, G., "Web Ontology Language (OWL) Reference Version 1.0," <http://www.w3.org/TR/owl-ref/>, 2002.

[17] Smith, M., McGuinness, D. and Welth, C., "Web Ontology Language (OWL) Guide Version 1.0," <http://www.w3.org/TR/owl-guide/>, 2002.

[18] Kim, W., Introduction to Object-Oriented Databases, The MIT Press, 1990.

[19] Choi, J.H., Yang, J.D. and Lee, D.G., "An object-based Approach to Managing Domain Specific Thesauri : Semiautomatic Thesaurus Construction and Query-based Browsing," International Journal of Software Engineering & Knowledge Engineering, Vol. 10, pp. 1-27, 2002.

[20] TouchGraph, <http://www.touchgraph.com>

[21] JENA 2 - A Semantic Web Framework, <http://www.hpl.hp.com/semweb/jena.htm>

김 기 헌

1997년 전북대학교 전자계산학과(학사) 1999년 전북대학교 전산통계학과(석사) 2001년 전북대학교 전산통계학과 박사수료. 2001년~현재 (주)케이테크 멀티미디어 DB연구소 선임연구원. 관심분야는 온톨로지, 바이오인포매틱스, 시멘틱 웹, 정보검색 등

최 재 훈

1994년 전북대학교 전자계산학과(학사) 1996년 전북대학교 전산통계학과(석사) 2000년 전북대학교 전산통계학과(박사) 2000년~현재 한국전자통신연구원 연구원. 관심분야는 바이오인포매틱스, 온톨로지, 시멘틱웹, OODBMS, 소프트웨어공학

양 재 동

1983년 서울대학교 컴퓨터공학과(학사) 1985년 한국과학기술원 전산학과(석사) 1991년 한국과학기술원 전산학과(박사) 1995년~1996년 Univer. of Florida, Visiting Scholar. 현재 전북대학교 전자정보공학부 교수. 관심분야는 멀티미디어 정보검색, 문서 정보 검색, 온톨로지, OODB, Expert System 등

박 천 수

1999년 충남대학교 컴퓨터과학과(학사) 2001년 충남대학교 컴퓨터과학과 컴퓨터과학전공(석사). 2001년~2003년 한국전자통신연구원 컴퓨터소프트웨어기술연구소 연구원. 2004년~현재 한국전자통신연구원 지능형로봇연구단 연구원. 관심분야는 지능형 로봇, 능동서비스 Planning, 웹 서비스