웹 문서의 의미적 연관성 기술을 위한 온톨로지 에디터

이 무 훈[†] · 조 현 규[††] · 조 현 성[†††] · 조 성 훈[††††] · 장 창 복[†††††] · 최 의 인[††††††]

요 약

웹의 확산과 더불어 웹상에 존재하는 정보의 양은 예측할 수 없을 정도로 증가하였고, 웹 사용자의 이용수준과 요구 사항도 매우 다양하고 복잡해졌다. 사용자가 원하는 정보와 의미적으로 정확히 일치하는 정보들을 검색하기 위해서는 웹 정보에 대한 정확한 의미 부여와 웹 정보 사이의 의미적 연관성을 기술할 수 있는 지식 표현 수단으로써 온톨로지가 필요하다. 이러한 필요성에 따라 W3C에서는 웹 자원에 대한 의미 표현 기술로 OWL(Web Ontology Language)이라는 웹 온톨로지 언어를 발표하였으나 아직 이를 효과적으로 생성, 편집할 수 있는 전용 에디터(editor)의 개발은 아직 미비한 실정이다. 따라서 본 논문에서는 웹 문서들 간의 의미적 연관성을 기술할 수 있는 OWL을 직관적인 인터페이스(interface)로 생성 및 편집할 수 있도록 OWL Parser, Internal DataModel, Visualization Module을 제공하는 온톨로지 에디터를 설계하고 구현하였다.

키워드 : 시맨틱 웹, 온톨로지, OWL, 온톨로지편집 도구

An Ontology Editor to describe the semantic association about Web Documents

Moo-Hun Lee[†] · Hynu-Kyu Cho[††] · Hyeon-Sung Cho[†††]
Sung-Hoon Cho[††††] · Chang-Bok Jang[†††††] · Eui-In Choi[††††††]

ABSTRACT

As the internet continues to grow, the quantity of information on the Web increases beyond measure. The internet users' abilities and requirements to use information also become varied and complicated. Ontology can describe correct meaning of web resource and relationships between web resources. And it can extract conformable information that a user wants. Accordingly, we need the ontology to represent knowledge. W3C announced OWL(Web Ontology Language), a meaning description technology for such web resources. But, the development of a professional use of tools that can compose and edit effectively is not yet developed adequately. In this paper, we design and implement an Ontology editor which generates and edits OWL documents through intuitional interface, with a OWL parser, a Internal DataModel, and a Serializer.

Key Words : Semantic Web, Ontology, Web Ontology Language, Ontology Editing Tool

1. 서 론

웹상에 존재하는 정보의 양은 예측할 수 없을 정도로 거대하다. 또한 웹의 확산으로 인하여 그 정보의 양은 기하급수적으로 증가하고 있다. 웹에는 단체와 개인들이 서로 다른 목적으로 생성한 수많은 문서와 정보가 포함되어 있다. 웹 사용자의 이용수준과 요구사항이 다양하고 복잡해짐에 따라 정확하고 적절한 정보를 제공하는 것이 중요한 문제가 되고 있다[1]. 이와 같은 문제점은 현재의 웹이 HTML(Hyper Text Markup Language) 기반의 표현 위주의 웹으로 구성되어 있기 때문이다. 이는 정해진 태그들만을 사용해야 한다는 점과 문서의 구조적인 정보를 표현할 수 없으므로 정보 추출이 어렵다는 단점을 가지고 있다. 이러한 HTML의 단점을 해결하기 위해 XML(eXtensible Markup Language)이 제안되었으나, XML은 단지 문법 기술에 초점을 두고 개발되었기 때문에 정보들 간의 관계를 표현하기에 부족하고 의미가 상속되지 못한다. 이러한 HTML과 XML의 단점을 보완하기 위해 RDF가 제안되었다[2]. RDF는 시맨틱 웹을 지원하기 위한 기반 구조이며, 분산된 다양한 자원들을 기술하기 위한 것으로 상이한 메타데이터를 효율적으로 교환하고 이해할 수 있는 정보 교환 수단을 제공한다. 하지만

† 준 회 원 : 한남대학교 컴퓨터공학과 박사과정
†† 정 회 원 : 한국전자통신연구원(ETRI)
††† 정 회 원 : 한국전자통신연구원(ETRI)
†††† 준 회 원 : 한남대학교 컴퓨터공학과 박사과정
††††† 준 회 원 : 한남대학교 컴퓨터공학과 박사과정
†††††† 종신회원 : 한남대학교 컴퓨터공학과 교수
논문접수 : 2005년 6월 11일, 심사완료 : 2005년 10월 20일

RDF는 정보간의 동치성과 같은 관계에 대한 표현 능력이 미약하고 추론을 위한 규칙을 표현할 수 없다는 단점을 가지고 있다. 이 문제점을 개선하기 위해 온톨로지, OIL(Ontology Inference Layer), DAML(DARPA Agent Markup Language)과 같은 언어가 개발되었다[3-5]. 이러한 기반 기술들을 토대로 기존 웹의 문제점들을 해결하기 위해 제안된 개념이 시맨틱 웹이다[5]. 시맨틱 웹은 컴퓨터 스스로가 웹에 연결된 정보의 의미를 인식하고 사용자가 요구하는 정보를 검색하여 검색된 정보에서 지식을 추론할 수 있는 기능을 제공한다. 그러나 앞에서 기술한 바와 같이 XML과 RDF는 컴퓨터가 XML 문서에 포함된 태그의 의미를 정확하게 표현하고 절차적 추론 과정을 수행할 수 있게 하기 위해서는 온톨로지의 개념이 필요하다. 온톨로지는 응용프로그램 사이에서 웹 기반 지식 처리, 공유, 재사용을 가능하게 하고, 전자상거래에서 구매자와 구입자간의 기계기반의 의사소통을 가능하도록 한다. 이와 같은 다양한 온톨로지 언어를 바탕으로 온톨로지 관련 연구가 진행 중이지만 이들은 각기 다른 온톨로지 언어를 기반으로 하고 있고, 각기 다른 방식으로 연구되었기 때문에 온톨로지 사이의 호환성에 대한 문제점을 가지고 있다[6]. 따라서 이를 해결하기 위해 W3C는 온톨로지 언어에 대한 표준화 작업이 진행하여 표준 온톨로지 언어인 OWL을 발표하였다. 이러한 상황에서 웹 자원에 의미 정보를 추가하기 위해서는 OWL에 대한 전반적인 지식 없이도 OWL 문서를 쉽게 생성할 수 있는 온톨로지 에디터가 필요하다. 국외에서 온톨로지 저작 도구에 대한 연구가 활발히 진행 중이지만, OWL을 위한 전용 도구가 미흡한 실정이며 기존의 도구들은 다양한 인터페이스를 제공하고 있지 못하여 온톨로지 전문가들조차도 사용하기 어렵다는 문제점을 가지고 있다. 따라서 본 논문에서는 이와 같은 문제점을 해결하기 위해 OWL 문서를 그래픽 인터페이스를 통하여 쉽게 생성 및 편집할 수 있고, 복잡한 OWL 문서를 효과적으로 분석할 수 있는 온톨로지 에디터를 설계하고 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 기존 온톨로지 저작 도구에 대해 살펴보고, 3장에서는 제안한 온톨로지 에디터의 설계 및 구현에 관하여 설명한다. 4장에서는 기존 온톨로지 편집 도구와 제안한 에디터를 비교분석하고, 5장에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련 연구

2.1 OilEd

영국의 맨체스터 대학에서는 DAML+OIL에 기반으로 트리(tree) 방식으로 온톨로지를 편집하는 도구인 OilEd를 개발하였다. OIL을 기반으로 하는 간단한 프리웨어 개발을 목표로 OilEd를 개발하였기 때문에 OilEd는 전반적인 온톨로지 구축 환경을 제공하지는 않는다. OilEd는 FaCT(Fast Classification of Terminologies) 추론 기관과 연결하여 DAML+OIL 온톨로지를 구축하는 과정에서 FaCT의 추론 기능을 활용하고 있다. DAML+OIL의 풍부한 표현력과 다양한 제약조건을 모두 지원하기 위해 노력한 OilEd는 시각적인 온톨로지 편집 환경을 제공하지 못함으로 대규모의 온톨로지 개발에 부적합하다[7-9].

2.2 Protégé OWL Plugin과 ezOWL Plugin

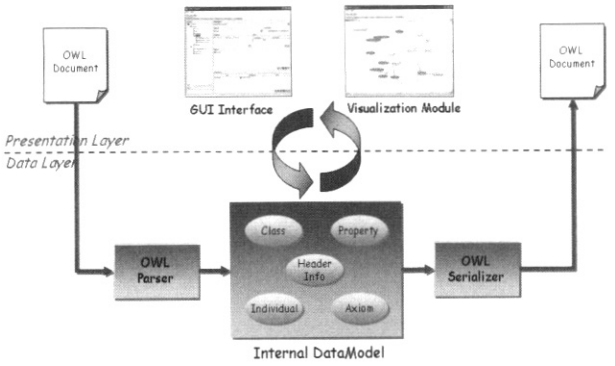
국립의료도서관(National Library of Medicine), NSF(National Science Foundation), DARPA (Defense Advanced Research Projects Agency)의 후원을 받아 개발된 Protégé는 스탠포드 의과대학의 의료정보학과(Stanford University School of Medicine, Stanford Medical Informatics)에서 지식기반의 구조를 작성하기 위해 15년간의 연구를 진행하였다. 따라서 Protégé는 현재 가장 광범위하게 활용되고 있는 온톨로지 개발/관리도구이며, 다른 지식 표현 언어와 호환이 가능하고, 온톨로지의 생성 및 수정을 위한 확장이 용이하다는 장점을 가지고 있다. 그리고 다양한 온톨로지 언어의 import/export 기능, 지식기반 데이터베이스 구축을 통한 질의 기능, 추론 기능 등의 풍부한 플러그인을 통한 확장성을 제공하고 있다. 최근 국내에서 개발된 ezOWL Plugin과 스탠포드에서 개발된 OWL Plugin은 온톨로지 언어의 표준으로 자리잡고 있는 OWL에 대한 편집 기능을 제공하고 있다[10]. 오랜 개발 기간을 통해 Protégé는 높은 완성도를 가지고 있지만, OWL이 표준화 되기 이전부터 개발되었던 도구이기 때문에 Protégé의 내부 온톨로지 모델은 OWL에서 제공하는 표준 온톨로지의 개념(concept)을 사용하지 않고, 기존에 Protégé 내에서 자체적으로 정의된 개념들을 사용하고 있다. 따라서 OWL을 위한 전용 도구로써 적합하지 않다[11, 12]. 그러므로 Protégé의 경우, 범용적으로 온톨로지를 개발/관리하는 시스템으로써 다양한 온톨로지 표현을 내부 온톨로지 모델로 변환하여 사용한다는 장점을 가지고 있으나, 표준 OWL을 작성하기 위한 전용 편집 도구로 활용하기에는 부적합한 실정이다.

2.3 OntoEdit

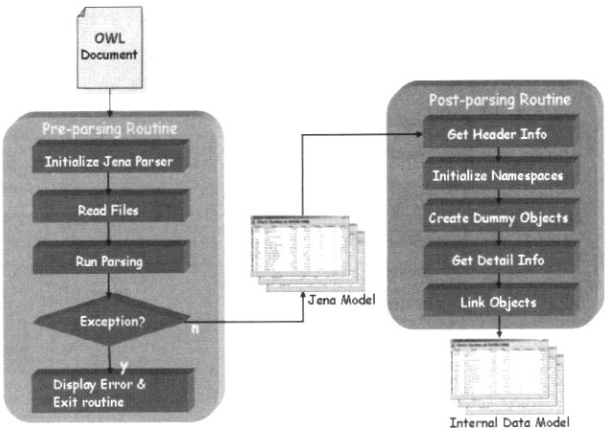
독일의 ontoprise® GmbH 사에서 개발한 OntoEdit은 사용자가 쉽게 온톨로지를 생성하고 편집할 수 있는 GUI 환경을 제공하고 있다. 또한 OntoEdit은 강력한 내부 온톨로지 모델을 이용하여 온톨로지 공학 환경(Ontology Engineering Environment)을 제공하고 있다. OntoEdit의 내부 모델은 개념(concept), 관계(relation), 원칙(axioms)이 가능한 중립적 모델링 표현 언어를 제공하고 있으며, 직관적인 인터페이스를 통하여 온톨로지에 대한 시각화 및 탐색을 지원하고 있다. 하지만 OntoEdit에서 사용하고 있는 내부 온톨로지 모델은 표현력에 있어 OWL을 충분히 지원하지 못하며, DAML+OIL에서 제공하지 못하고 OWL에서만 제공하고 있는 제약조건 및 동치관계를 명시할 수 없다는 단점을 가지고 있다[13, 14].

3. 온톨로지 에디터의 구조

본 논문에서 W3C를 중심으로 온톨로지의 표준화를 위해



(그림 1) 온톨로지 에디터의 시스템 구조



(그림 2) 파싱 처리 루틴

제안된 웹 온톨로지 언어인 OWL의 표현력을 전부 수용하고, OWL의 생성/편집을 용이하기 위하여 Internal DataModel을 제안한다. 내부 데이터 모델을 이용하여 OWL을 생성/편집하기 위한 온톨로지 에디터의 전체 시스템 구조는 (그림 1)과 같다. 설계한 온톨로지 에디터는 각 편집 단계에서 적용되는 모든 사항을 Internal DataModel에 반영할 수 있도록 크게 Class, Property, Individual Axiom, Header Info의 다섯 가지 클래스로 구성하였다. 이를 통하여 사용자 인터페이스 상에서 편집 된 내용을 각 해당 클래스(class)에 반영하고 이를 Visualization Module이 OWL 문서에 맞게 작성하여 출력하여 준다.

온톨로지 에디터는 크게 표현 계층(presentation layer)과 데이터 계층(data layer)으로 구분된다. 표현 계층은 실제 OWL 문서를 편집하고 분석, 탐색하는 사용자 인터페이스 부분이며, 데이터 계층은 읽어 들인 OWL 문서나 편집 중인 OWL 문서의 상태를 반영하게 된다. 텍스트 파일 형태의 OWL 문서를 입력 받아 Internal DataModel로 변환하는 OWL Parser, 온톨로지 의미 구성 요소들의 정보를 저장하기 위한 Internal DataModel, 생성된 모델을 사용자에게 시각적으로 표시하고 생성할 수 있도록 하기 위한 사용자 인터페이스, 구축된 온톨로지의 정보들을 시각화하여 확인하고 탐색할 수 있도록 하기 위한 Visualization Module, 그리고 최종적으로 생성된 정보들을 바탕으로 다시 OWL 문서를 생성해 주는 OWL Serializer로 구성된다.

3.1 OWL Parser

OWL 문서의 구문을 검사하고, OWL 문서에 포함된 의미표현 요소들을 추출하는 OWL Parser는 외부 OWL 문서를 읽어들여 Internal DataModel을 생성한다. 파싱(parsing) 과정은 우선 OWL 문서를 입력받아 Jena 온톨로지 모델을 구축한 후, 문서의 각 의미 요소마다 Jena 모델을 Internal DataModel로 변환한다.

파싱 처리 과정은 Jena Model을 생성하기 위한 pre-parsing 단계와 Internal DataModel을 생성하기 위한 post-parsing 단계로 구성하였다. pre-parsing 단계에서는 입력된 텍스트 형태의 OWL 문서를 Jena API를 이용하여 내부 Jena Ontology 모델로 변환한다[15]. 생성된 내부 Jena Ontology 모델은 post-parsing 단계를 거쳐 실제 시스템에서 사용될 Internal DataModel로 변환한다. 변환 단계는 먼저 온톨로지의 헤더 정보를 확인하여 온톨로지에서 사용된 네임스페이스(namespace)를 내부 모델에 등록하고, 클래스, 속성(property), 개체(individual)에 대한 더미 객체(dummy object)를 생성한다. 더미 객체가 생성된 후에는 모든 객체에 대해 해당 객체가 갖고 있는 세부 의미 요소들을 추출하고 내부 모델에 반영한다. (그림 2)는 OWL Parser의 처리 단계를 보여주고 있다.

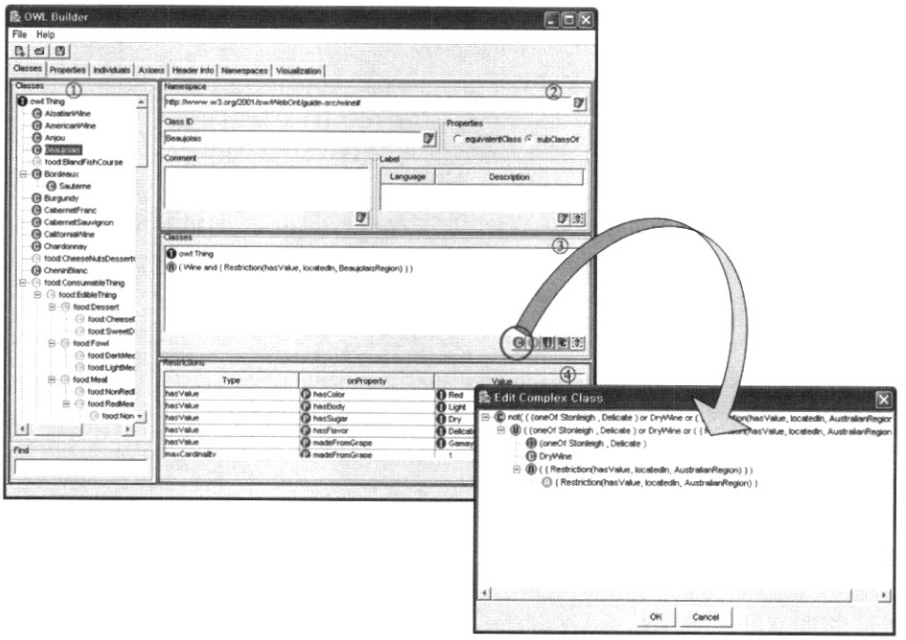
3.2 사용자 인터페이스

기존 온톨로지 편집 도구는 OWL의 다양한 표현을 전부 만족시키지 못하고 있지만 제안한 온톨로지 에디터의 사용자 인터페이스는 OWL 문서에서 표현하고 있는 다양한 의미 표현을 모두 하고 있다. 따라서 본 논문에서는 OWL 명세서 자세한 분석을 통해 이를 효과적으로 편집할 수 있도록 6개의 하위 모듈로 나누어 설계하였다. 사용자 인터페이스를 구성하는 6개 하위 모듈은 기본적인 OWL 요소를 정의하는 클래스, 속성, 개체 모듈, 하나의 클래스나 개체 상에서 정의할 수 없는 DisjointWith, AllDifferent 요소를 정의하는 axiom 모듈, OWL 온톨로지의 헤더 정보인 versionInfo, priorVersion, import, backCompatibleWith, inCompatibleWith를 정의하는 헤더(header), 그리고 온톨로지 내부에서 사용하는 네임스페이스에 대해 수정을 제공하는 네임스페이스 모듈로 구성된다. 다음의 <표 1>은 OWL의 기본 요소를 정의하는 클래스, 속성, 개체의 세부 지원 기능을 나타내고 있다.

클래스 정의 인터페이스에서는 클래스를 크게 named 클래스, anonymous 클래스, one-of, union 클래스, intersection 클래스, complement 클래스로 구분하였다. 일반적인 named 클래스를 정의하는 사용자 인터페이스는 (그림 3)과 같다. ① 클래스 트리(tree), ② annotation 정의부, ③ 슈퍼 클래스 정의부, ④ restriction 정의부로 구성하였으며, ①은 온톨로지에 정의된 모든 클래스들의 서브클래스(sub class) 관계를 트리 형태로 표시한다. 선택된 클래스에 대해 우측

〈표 1〉 하위 사용자 인터페이스의 지원 기능

클래스	속성	개체
• Annotation 정의 - ID, Label, Comment • subClassOf • equivalentClass • Class Combination - intersectionOf - unionOf - complementOf • Restriction 정의 - Cardinality - Property Restriction	• Annotation 정의 - ID, Label, Comment • Property 특성 정의 - Functional - inverseFunctional - Symmetric - Transitive • subPropertyOf • equivalentProperty • inverseProperty • Domain • Range	• Annotation 정의 - ID, Label, Comment • Individual간 관계 - sameIndividualAs - differentFrom • Relation 정의



(그림 3) named Class와 complex Class 편집 인터페이스

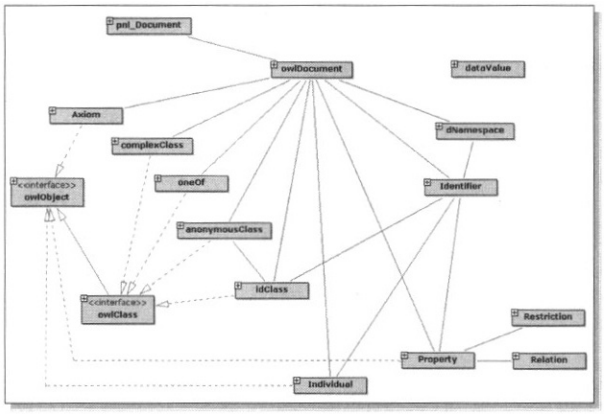
세부 정보 패널(②, ③, ④)에서 수정할 수 있도록 구성하였다. ②에서는 해당 클래스에 대한 메타데이터(네임스페이스, id, comment, label)를 정의할 수 있도록 하였다. ③은 선택된 클래스의 슈퍼 클래스(혹은 equivalent 클래스)를 정의하며, 다양한 형태의 anonymous 클래스를 정의 할 수 있다. ④는 restriction을 정의하는 부분으로 Cardinality, maxCardinality, minCardinality 등의 cardinality 정보와 Allvalues From, someValuesFrom, hasValue 등의 속성 restriction을 정의할 수 있도록 구성하였다.

(그림 3)에서와 같이 union 클래스, intersection 클래스, complement 클래스 등의 complex 클래스 편집 시에는 complex 클래스 편집 인터페이스를 통하여 생성/편집하게 된다. Complex 클래스 편집 인터페이스는 트리 형태로 내부 요소들을 표현하여 세부 요소들의 편집이 용이하도록 구성하였다.

3.3 Internal DataModel

파싱 단계에서 추출된 요소들은 Internal DataModel로 변환하여 메모리에 저장함으로써 기존 편집 도구보다 신속하고 빠른 참조를 통해 효율적으로 OWL 문서를 처리하고 편집 환경에서 즉각 내부 모델에 반영하도록 설계하였다. Internal DataModel은 OWL 스펙에서 지원하고 있는 다양한 의미적 관계와 제약 조건을 충실하게 제공할 수 있도록 최상위의 owlDocument를 비롯하여 owlObject, owlClass, idClass, anonymousClass, oneOf, complexClass, Axiom, Property, Individual, dNamespace, dataValue, Identifier, Restriction, Relation의 15개의 클래스로 정의하였다. (그림 4)는 각각의 클래스들 간의 관계를 보여주는 Internal DataModel의 클래스 다이어그램을 보여주고 있다.

owlDocument는 내부 데이터 모델의 최상위 객체로 온톨로지의 모든 의미요소들을 포함하며, 클래스, 속성, 개체 등 자원 식별자를 통해 접근하는 객체에 대한 해쉬 테이블(hash table)을 유지함으로써 보다 빠른 객체 참조가 가능하며, 또한 complex 클래스 편집 화면과 같이 추가적으로 사용되는 사용자 인터페이스에서 빈번하게 이용되는 클래스와



(그림 4) 내부 데이터 모델의 클래스 다이어그램

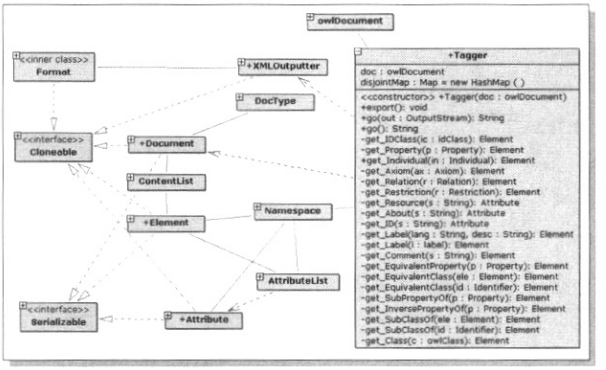
속성의 트리 모델도 Internal DataModel에서 관리하도록 하였다. 논문에서 제안하고 있는 내부 모델에 정의된 각각의 객체는 OWL 스펙에서 정의한 의미 요소들과 1:1로 대응되도록 설계하였으며, 의미 요소들을 저장하기 위한 멤버 변수들과 이들을 조작하기 위한 메소드(method), 그리고 사용자 인터페이스를 통해 발생된 이벤트(event)를 처리하기 위한 메소드로 구성되어 있다.

3.4 Visualization Module

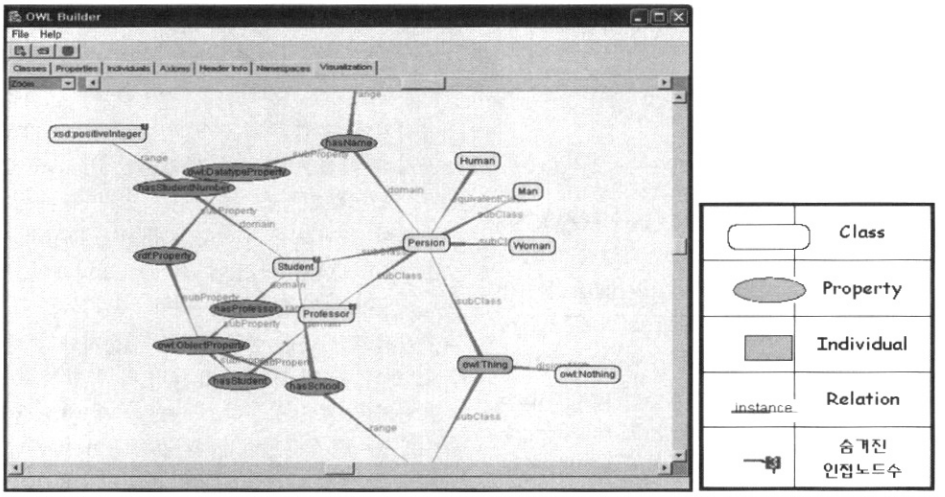
온톨로지 에디터의 Visualization Module은 구축된 의미 정보들을 직관적으로 판독할 수 있도록 노드(node)와 아크(arc) 형태의 그래프로 표현하고 객체들을 선택하여 탐색할 수 있는 기능을 제공해준다. 따라서 단순 GUI 환경의 기존 편집 도구보다 효율적으로 온톨로지를 관리하고 분석할 수 있는 장점이 있다. Visualization Module에서 온톨로지의 의미 요소인 클래스, 속성, 개체를 각각의 노드로 표현하며, 이들 간의 관계는 아크로 표현한다. Visualization Module은 Graph Layout API를 응용하여 온톨로지의 의미 구조를 효과적으로 도식화하였다[16]. Visualization 탭이 선택되면 온톨로지에 구축된 의미 요소들을 이용하여 그래프를 초기화하며, 탐색 과정에서 선택된 노드와 인접한 노드를 중심으로 화면을 재구성하여 표시하도록 구성하였다. 또한 탐색 과정에서 자유로운 zoom-in, zoom-out이 가능하도록 하였고, 전체 노드의 수에 따라 사용자가 직접 선택 노드와 몇 단계까지 연결된 노드를 표시할 것인지에 대한 범위를 선택할 수 있어 편집중인 온톨로지의 단순 의미 구조와 세부 의미 구조를 파악함으로써 효과적으로 온톨로지를 편집할 수 있다. (그림 5)는 Visualization Module의 실행 화면과 노드와 아크에 대한 도식을 설명하고 있다.

3.5 OWL Serializer

OWL Serializer는 사용자 인터페이스에서 생성/편집되어진 Internal DataModel을 다시 텍스트 형식의 OWL 문서로 변환한다. OWL Serializer는 Internal DataModel의 각 요소들을 XML-DOM 모델로 변환한다. 그리고 모든 요소들이 XML-DOM 모델로 변환되면, XMLOutputter 객체를 이용하여 최종적으로 OWL의 문법에 맞는 텍스트 형식의 OWL 문서를 출력해 준다. (그림 6)은 OWL Serializer의 클래스 다이어그램을 보여주고 있다.



(그림 6) 직렬화 모듈의 클래스 다이어그램



(그림 5) 시각화 및 탐색 모듈의 실행 화면

(그림 7) 직렬화 모듈의 처리 결과

〈표 2〉 기존 도구와의 비교 분석

	OilEd	Protégé OWL Plugin	Protégé ezOWL Plugin	OntoEdit	제안한 온톨로지 에디터
온톨로지 데이터 모델	DAML+OIL spec.	기존 Protégé 데이터 모델	기존 Protégé 데이터 모델	자체 온톨로지	OWL spec.
표현 온톨로지	DAML+OIL	OWL	OWL	자체 온톨로지	OWL
호환성	하	상	상	하	상
표현력	상	중	중	중	상
데이터 질의	×	○	○	×	×
시각화	×	×	○	○	○
탐색 기능	×	×	×	○	○

OWL Serializer은 내부 모델의 클래스, 속성, 개체 등 각각의 요소들에 대해 XML 형태로 변화하기 위해 메소드를 제공하고 있으며, 사용자의 이벤트에 의해 수정된 내부 모델을 저장하기 위해서는 OWL Serializer을 통해 텍스트 형태의 OWL 문서로 변환해야 한다. OWL Serializer은 매핑 규칙을 통해 내부 데이터 모델을 XML-DOM 모델로 변환한다. (그림 7)에서는 OWL Serializer에 의해 변환하여 생성한 최종 OWL 문서가 출력된다.

4. 비교 분석

본 논문에서 제안하고 있는 온톨로지 에디터는 OWL 문서의 모든 문법을 모두 표현하고 효과적으로 처리하기 위하여 OWL Parser, 사용자 인터페이스, OWL Serializer를 제공하고 있다. 특히 OWL의 다양한 표현력과 의미 구조, 제약 조건을 반영하는 Internal DataModel과 구축된 온톨로지를 쉽고 직관적으로 관리, 분석 할 수 있는 Visualization Module을 제공하고 있다. 이러한 OWL 편집 도구의 기능을 바탕으로 기존 온톨로지 편집 도구와 비교 분석하였다. <표 2>는 OWL 편집과 온톨로지 관리에 필요한 기능 명세와 그 기능 지원 여부를 정리한 것이다. Protégé의 경우, 처음 의료분야의 지식 정보에 대한 온톨로지를 구축하기 위하여 연구를 시작한 온톨로지 저작 시스템으로 개발 초기에는 자체 모델을 사용하여 호환성이 떨어진다는 단점이 있었으나, 현재는 다양한 플러그인을 통하여 확장성을 제공하고 있다. 하지만 이러한 플러그인을 통한 확장성은 Protégé에서 사용하고 있는 내부 데이터 모델을 기반으로 하고 있고 Protégé의 온톨로지 모델에 대한 전문지식을 가지고 있어야 이를 활용할 수 있기 때문에 OWL을 위한 전용 편집 도구로 활용하기에 부적절하다. 또한 Protégé ezOWL Plugin의 경우 시각화 기능을 제공하고 있지만 Protégé OWL Plugin의 경우 시각화 기능을 제공하고 있지 않아서 OWL의 편집 및 분석이 용이하지 못하다. OilEd는 DAML+OIL을 기반을 두고 있어 DAML+OIL에 적합한 인터페이스로 고정되어져 있다. 그리고 온톨로지에 대한 시각화를 제공하고 있지 않아 복잡하고 거대한 온톨로지를 생성하는데 많은 어려움을 가지고 있다. OWL과 DAML+OIL의 문법이 유사하지만, OWL의 표현력이 훨씬 풍부하고 다양한 제약 조건을 지정할 수 있어 OilEd를 확장하는데 많은 문제점이 따른다. 하지만 본 논문에서 제안한 온톨로지 에디터는 온톨로지의 새로운 표준인 OWL을 위한 전용 편집 도구로써 효율적으로 온톨로지를 생

성/편집할 수 있다. 또한 시각화 및 탐색 모듈을 지원함으로써 복잡한 온톨로지를 효과적으로 분석할 수 있다.

5. 결 론

본 논문에서는 OWL 기반의 온톨로지 문서를 파싱하여 Internal DataModel로 구축하고, 시각적인 사용자 인터페이스를 통해서 효율적으로 OWL을 생성/편집할 수 있는 온톨로지 에디터를 설계 및 구현하였다. 제안된 에디터는 OWL에서 지원하는 모든 표현들을 수용하기 위해 클래스, 속성, 개체, axiom, 헤더 정보, 네임스페이스에 따른 분리된 인터페이스를 제공하며, 이를 위해 각각에 대한 Internal Data Model을 설계하고 구현하였다. 온톨로지 에디터에서 사용하고 있는 내부 모델은 W3C의 온톨로지 표현 언어인 OWL 스펙의 분석을 통하여 설계하였기 때문에 풍부한 표현력과 호환성을 제공하는 장점을 가지고 있다. 또한 본 논문에서 제안한 그래프 기반의 Visualization Module은 구축된 의미 요소들을 노드 앤 아크 다이어그램으로 표현함으로써 요소들 간의 관계를 보다 직관적으로 분석할 수 있으며, 탐색기능을 통해 의미들을 추적함으로써 보다 효율적인 온톨로지 관리가 가능하다. 그리고 W3C의 표준인 OWL을 기반으로 하였기 때문에 시맨틱 웹뿐만 아니라 다양한 온톨로지 분야에서 의미적 연관성을 기술하는 OWL 문서 작성에 폭넓게 활용할 수 있을 것이다.

향후 연구과제로는 다양한 분야에서 서로 다른 언어로 작성된 온톨로지를 OWL로 변환하여 통합 할 수 있는 방안에 대한 연구가 진행되어야 할 것이다.

참 고 문 헌

[1] 이재호, "시맨틱 웹의 온톨로지 언어", 한국정보과학회 정보과학회지, 제21권 제3호 pp.18-27, 2003

[2] 김홍기, 김학래, 이강찬, 정지훈, 이재호 외, "웰드와이드웹에서 시맨틱 웹으로", 마이크로소프트웨어 시맨틱 웹 특집, pp. 242-301, 2002

[3] Berners-Lee, t., et al., "The Semantic Web," Scientific American, 2001.

[4] Comez-Perez, a. and Corcho, O., "Ontology languages for the Semantic Web," IEEE Intellignet Systems, Vol.17, No.1, pp.54-60, 2002

[5] Frank van Harmelen, Peter F. Patel-Schneider, and Ian Horrocks. Reference Description of the DAML+OIL (March 2001) Ontology Markuk Language. DAML +OIL Document, URL http://www.daml.org/2000/12/reference.html, March. 2001.

[6] Jeff Z. Pan, Ian Horrocks, "Metamodeling Architecture of Web Ontology Languages," In Proceedings of the Semantic Web Working Symposium, pages 131-149, Stanford, July, 2001.

[7] D. Fensel, F. Harmelen, M. Klein, H. Akkermans, "On-To-Knowledge : Ontology-based Tools for Knowledge Management," Proceeding of eBusiness and eWork 2000 Conference (EMMSEC 2000), 2000.

[8] OilEd, http://oiled.man.ac.uk/

[9] D. Fensel, et al., "Semantic Web Application Areas," the 7th International Workshop on Applications of Natural Languages to Information system, Stockholm, Sweden, June, 2002.

[10] Asunción Gómez-Pérez, Oscar Corcho, "Deliverable 1.3 : A survey on ontology tools," Onto-Web, May, 2002.

[11] Protégé OWL Plugin, http://protege.stanford.edu/plugins/owl/

[12] ezOWL, http://iweb.etri.re.kr/ezowl/

[13] OntoEdit, http://www.ontoprise.de/products/ontoedit_en

[14] York Sure, et al., "OntoEdit : Collaborative Ontology Development for the Semantic Web," Proceedings of the First International Semantic Web Conference on The Semantic Web, p.221-235, June, 09-12, 2002.

[15] Jena, http://www.hpl.hp.com/semweb/jena2.htm

[16] TouchGraph, http://www.touchgraph.com/

이 무 훈

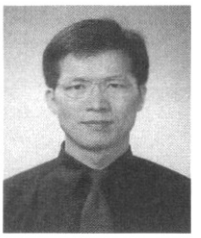
email : mhlee@dblab.hannam.ac.kr

2002년 한남대학교 컴퓨터공학과(공학사)

2004년 한남대학교 컴퓨터공학과(공학석사)

2004년~현재 한남대학교 컴퓨터공학과 박사과정

관심분야 : Semantic Web, Web Service, Web search engine, Data Stream Management System



조 현 규

email : hkcho@etri.re.kr

1988년 한국외국어대학교(학사)

1990년 고려대학교 대학원(경영학석사)

1997년 한남대학교 대학원(경영학박사)

1990년~현재 한국전자통신연구원(ETRI) 지능형로봇연구단 지능형서비스플랫폼연구팀 책임연구원

관심분야 : Semantic Web, ebXML, Web Service, Intelligent Robot, Ubiquitous Computing

조 현 성

email : hsc@etri.re.kr

1995년 충남대학교 컴퓨터공학과(공학사)

1997년 충남대학교 컴퓨터공학과(공학석사)

1997년~현재 한국전자통신연구원(ETRI) 지능형로봇연구단 지능형서비스플랫폼연구팀 선임연구원

관심분야 : Semantic Web, Web Service, Web Search, Intelligent Robot, Ubiquitous Computing

조 성 훈

email : shcho@dblab.hannam.ac.kr
2001년 한남대학교 컴퓨터공학과(공학사)
2003년 한남대학교 컴퓨터공학과(공학석사)
2004년~현재 한남대학교 컴퓨터공학과 박사과정
관심분야 : RDF, Semantic Web, Ontology, ebXML, Web Service

장 창 복

email : chbjang@dblab.hannam.ac.kr
2000년 한남대학교 컴퓨터공학과(공학사)
2002년 한남대학교 컴퓨터공학과(공학석사)
2002년~현재 한남대학교 컴퓨터공학과 박사과정
관심분야 : Middleware, Active network, Context Modeling, Semantic Web, Software Security, OVPN

최 의 인

email : eichoi@dblab.hannam.ac.kr
1982년 한남대학교 계산통계학과(학사)
1984년 홍익대학교 전자계산학과(이학석사)
1995년 홍익대학교 전자계산학과(이학박사)
1985년~1988년 공군 교육사 전산실장
1992년~1996년 명지전문대학 전자계산과 조교수
1996년~현재 한남대학교 컴퓨터공학과 교수
2003년 UCLA visiting scholar
관심분야 : Web Service, Semantic Web, Web search engine, Ubiquitous Computing, Context-Aware Retrieval, Context Modeling, Data Stream Management System