

# 온톨로지 생성과 공유를 위한 시맨틱 웹 기반 위키 시스템

(A Semantic Web-enabled Wiki System for Ontology Construction and Sharing)

김 현 주 <sup>†</sup>    최 중 민 <sup>††</sup>  
(Hyunjoo Kim)    (Joongmin Choi)

**요 약** 시맨틱 웹은 컴퓨터가 처리 가능한 의미 정보를 표현하고 공유할 수 있는 보편적인 매체를 개발하는 것이 목적이며, 따라서 시맨틱 웹에서는 이러한 의미 정보를 표현하는 온톨로지를 웹에 배포하여 이용 가능하게 만드는 것이 매우 중요하다. 하지만 현존하는 대부분의 온톨로지 저작 도구는 웹이 아닌 환경에서 운용되기 때문에 작성된 온톨로지를 바로 웹에 배포할 수 없으며 온톨로지를 여러 사람이 공동으로 저작할 수 없다는 단점이 있다. 이 논문에서는 위키(Wiki)를 이용하여 온톨로지를 쉽게 웹에 배포할 수 있고 온톨로지 생성과 공유를 용이하게 해주는 기반구조를 제안한다. 위키는 사람을 대상으로 하는 지식 공동 저작과 공유를 위한 기반 구조 중 하나로서 웹에서 운용되며, 위키의 내용은 웹 문서 서식을 위한 간단한 마크업 언어와 자연어로 구성된다. 이 논문은 보다 용이한 온톨로지 생성과 공유를 위해 기존의 위키 시스템에 시맨틱 웹 요소를 추가하여 인간을 위한 지식 공동 저작과 공유를 위한 시스템인 동시에 에이전트 소프트웨어도 쉽게 접근하여 온톨로지 정보를 얻을 수 있는 시맨틱 웹 기반 구조를 제안하였으며 이를 통해 시맨틱 조회, 시맨틱 탐색, 시맨틱 질의를 실현하고자 하였다.

**키워드** : 시맨틱 웹, 온톨로지, 정보공유, 위키, 시맨틱 위키

**Abstract** The Semantic Web has the objective of developing universal media in which machine-processable semantic information can be represented and shared, and it is therefore important to distribute ontologies that represent this kind of semantic information to the Web and make them available to multiple parties. However, the current ontology authoring tools are not operating on the Web, which makes it difficult to distribute ontologies directly to the Web and to create and edit them collaboratively with other people. This paper proposes a framework that facilitates the ontology construction and sharing, realizing easy distribution of ontologies to the Web. Wiki is one of the frameworks for collaborative construction and sharing of knowledge on the Web, and Wiki contents consist of natural language texts and simple markup language for visualization. For better collaboration in creating and sharing ontologies, this paper suggests the Semantic Wiki that embodies the Semantic Web features to the existing Wiki system. The Semantic Wiki framework facilitates the collaboration in ontology co-authoring and sharing for people, and at the same time, makes it possible for the agent software to easily manage the ontology information. Eventually, the Semantic Wiki system accomplishes various tasks including the semantic view, the semantic navigation, and the semantic query.

**Key words** : Semantic Web, Ontology, Information Sharing, Wiki, Semantic Wiki

## 1. 서 론

시맨틱 웹은 HTML 언어를 기반으로 한 기존의 웹의 한계점을 극복하고 컴퓨터가 처리 가능한 의미 정보를 표현할 수 있도록 보편적인 매체를 개발하고자 하는 프로젝트이다[1,2]. 시맨틱 웹에서 온톨로지는 의미 정보를 나타내기 위해 필요한 개념(concept)과 관계(relation-

· 이 논문은 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(과제번호 R05-2003-000-10223-0)

† 학생회원 : 한양대학교 컴퓨터공학과  
hjkim@cse.hanyang.ac.kr

†† 종신회원 : 한양대학교 컴퓨터공학과 교수  
jmchoi@cse.hanyang.ac.kr

논문접수 : 2005년 6월 21일  
심사완료 : 2006년 6월 2일

ship)를 정의한 정형적 명세(formal specification)이다[3]. 시맨틱 웹의 온톨로지를 기술하는 언어로 RDF와 OWL이 있으며, 온톨로지는 (Subject, Predicate, Object) 또는 (Resource, Property, Value)와 같은 트리플 단위로 나타낼 수 있다[4,5]. 트리플 구조는 Subject와 Object를 그래프의 노드(node)로, Predicate을 레이블이 있는 그래프의 아크(arc)로 묘사하여 그래프 구조로 나타낼 수 있다. 하지만, 이러한 언어를 사용하여 온톨로지를 웹에 배포하려면 시맨틱 웹에 대한 지식 외에도 XML과 웹에 대한 사전 지식을 요구한다. 현재 Protégé나 OntoEdit와 같은 뛰어난 온톨로지 저작 도구가 존재하나, 웹이 아닌 환경에서 동작하고, 여러 명이 공동으로 온톨로지를 저작할 수 없으며 이러한 도구를 사용하여 온톨로지를 직접 웹에 배포할 수 없다는 단점이 있다.

시맨틱 웹 이전에 제안된, 사람이 웹에서 지식을 공동 저작하고 공유하기 위한 기반 구조로 위키(Wiki)가 있다[6]. 위키는 하나의 웹 사이트로서 문서 서식을 위한 간단한 마크업 언어와 자연어로 작성된 텍스트를 저장한다. 위키에 접근하는 사람이면 누구나 위키의 내용을 자유롭게 추가하고 수정할 수 있다. 위키는 웹에서 자연어로 표현 가능한 지식을 저장하는데 용이한 기반구조이다. 위키의 페이지는 HTML의 하이퍼 링크와 유사한 링크로 서로 연결되며, 위키 전체를 페이지가 노드이고 링크가 아크인 그래프 구조로 나타낼 수 있다.

이 논문은 시맨틱 웹의 온톨로지와 위키 시스템을 그래프 구조로 나타낼 수 있다는 점에 착안하여 기존의 위키 시스템에 시맨틱 웹 요소를 추가하여 시맨틱 웹 문서를 웹에서 자유롭게 저작하고 공유할 수 있는 웹 기반구조를 제안하였다. 이 기반구조에서는 자연어로 표현된 지식과 함께 시맨틱 웹 언어를 위한 메타 데이터를 저장하고 처리할 수 있다. 시맨틱 웹에 대한 관심이 고조되는 가운데 시맨틱 웹을 이용한 지식 정보 저장소의 모델은 현재의 시맨틱 웹의 연구 방향에 새로운 전환점을 제시하는 사례가 될 수 있을 것이다[7].

## 2. 관련연구

‘위키’는 ‘빨리’를 뜻하는 하와이 원주민어로, 빠르게 배우고 사용할 수 있는 특성을 나타낸다. 위키는 사용자가 위키의 어느 한 페이지가 구성 또는 내용이 잘못되어 있다고 생각되면 원래 그 페이지의 저자가 아니더라도 누구나 그것을 수정하여 바로 잡을 수 있도록 하는 최초 설계 원칙을 잘 반영하고 있다[8]. 이러한 원칙은 점진적으로 위키의 내용을 진화하는 방향으로 발전시키고 신뢰할 수 있도록 해준다.

위키의 내용이 개선되는 방향으로 진화하는 것과 같이, 위키의 구조도 자유롭게 진화하는 방향으로 발전한

다. 일반적인 웹 사이트는 기획 단계에서 미리 설계된 정적인 구조를 따라서 구성된다. 이와 달리 위키는 페이지의 생성과 갱신이 자유로우며 시간이 지날수록 진화하는 그래프 구조로 볼 수 있다. 페이지와 링크는 사용자가 자유롭게 추가와 삭제할 수 있기 때문에 시간이 지남에 따라 위키의 구조가 동적으로 바뀐다. 위키의 내용은 그림 1에 나타낸 것과 같이 일반 텍스트와 HTML에 비하여 단순한 마크업 언어로 구성되어 있다. 이것을 이용하여 시각적인 표현을 쉽게 할 수 있고 페이지의 생성과 수정을 편리하게 할 수 있다.

이러한 위키에 시맨틱 웹 요소를 추가하려는 시도가 이전에도 있었는데 그 중 하나가 Platypus 위키이다[9]. Platypus 위키의 기본 구조는 일반적인 위키의 구조와 동일하며, 차이점은 기존의 위키에서 사용하던 일반 위

```

-- Do as you like --
'''Ambiguous but yet pretty good working example'''
'italics'; 'bold'; 'bold italics';
'mixed 'bold' and italics';

* Item 1
* Item 2
* Item 3
* Item 4
* FrontPage - Camel Case Link
* RecentChanges shows 'latest' updated pages.

```

(a) 위키 마크업 언어

```

<p><h3>Do as you like </h3>
</p><p>
<b>Ambiguous but yet pretty good working example</b>
</p><p>
<i>italics</i>; <b>bold</b>; <b><i>bold italics</i></b>
<i>italics</i></b></i>; <i>mixed <b>bold</b> and italics</i>;
<hr />
<li>Item 1</li>
<li>Item 2</li>
<li>Item 3</li>
<li>Item 4</li>
<li><a href="FrontPage">FrontPage</a> - Camel Case
Link</li>
<li><a href="RecentChanges">RecentChanges</a> shows
<i>latest</i> updated pages.</li></p>

```

(b) HTML로 변환된 내용

## Do as you like

**Ambiguous but yet pretty good working example**

*italics*; **bold**; ***bold italics***; ***mixed bold and italics***;

- Item 1
- Item 2
- Item 3
- Item 4
- [FrontPage - Camel Case Link](#)
- [RecentChanges](#) shows *latest* updated pages.

(c) 브라우저에 표시되는 내용

그림 1 위키 페이지와 마크업 언어

키 페이지와 RDF/XML 데이터를 가지고 있는 시맨틱 웹 페이지가 분리된 형태로 공존하며 편집도 독립된 다른 창을 통하여 행해진다는 것이다. 일반 위키 페이지에서 `rdfs:Class`와 같은 형식으로 주소 공간(namespace)과 자원(resource)의 이름을 표기하면 다른 페이지로 접근하는 새로운 링크로 해석한다. 그러나 일반 페이지와 RDF/XML 페이지가 완벽하게 독립적으로 존재하기 때문에 일반 위키 편집창에서 `rdfs:Class`라고 기록한다고 해서 자동으로 RDF/XML 페이지에 메타 데이터가 갱신된다거나 혹은 그 반대로 RDF/XML 편집창에서 새로운 Class를 선언했다고 해서 일반 페이지에 그에 대한 사항이 자동으로 나타나지 않는다. 그리고 탐색이 일반 위키 페이지의 링크를 중심으로 이루어지기 때문에 일반 위키 페이지에서 다른 페이지로 접근하는 링크가 사라지게 될 경우, 링크가 사라진 다른 페이지에 접근하여 RDF/XML 온톨로지 데이터를 조회를 하려면 여러 단계를 거쳐야 한다. 본 논문은 Platypus 위키의 단점을 보완한 개선된 시맨틱 위키를 제안한다.

참고로 시맨틱 위키와 유사하게 웹에서 온톨로지를 자유롭게 생성하고 공유할 수 있는 프로그램으로 시맨틱 블로거(Semantic Blogger)가 있다[10]. 블로그(Blog)는 개인 사용자가 웹에 시간 순으로 글을 출판할 수 있는 시스템이다. 게시물을 분류별, 시간별로 조회할 수 있으며 표준화된 XML/RPC 블로그 API를 제공한다. 이 API를 이용하여 다른 블로그의 글에 대한 주석을 자신의 블로그에 적을 수 있고, 주석의 대상인 글이 있는 블로그는 이 사실을 알 수 있다. 블로그는 매우 단순한 구조와 사용법을 제공하기 때문에 컴퓨터와 웹에 대한 사전 지식이 없어도 누구나 손쉽게 사용할 수 있으며, 전 세계적으로 개인 홈페이지에 탑재되어 활발하게 사용되고 있다. 시맨틱 블로거는 일반 블로그에 시맨틱 웹 요소를 결합한 것이다. 시맨틱 블로거의 글은 RDF 형태로 저장된다. RDF로 저장되는 내용은 글의 제목, 분류, 작성자, 작성 시각, 내용, 글의 URI, 고유번호이다. 이렇게 저장된 의미정보를 담고있는 메타데이터와 온톨로지를 이용하여 기존 블로그에서 불가능했던 시맨틱 조회(Semantic View), 시맨틱 탐색(Semantic Navigation), 시맨틱 질의(Semantic Query)가 가능해졌다.

### 3. 시맨틱 위키의 설계

#### 3.1 시맨틱 위키의 특징

먼저 OWL로 표현된 자원과 위키 페이지를 표현하는 방법에 대하여 고려해야 한다. Platypus 위키에서는 OWL 클래스와 속성 등 모든 자원에 1:1로 대응하도록 위키 페이지와 온톨로지를 분리된 형태로 저장하였다. 여기서 탐색의 기준이 되는 링크는 위키 페이지의 링크이다. 이

방법은 위키 페이지의 URL과 OWL 자원의 URI가 일치하고, OWL 자원에 대한 속성이 바로 위키 페이지 사이의 링크가 될 수 있다는 장점이 있지만 현실적으로 표현하고자 하는 온톨로지가 복잡해질 경우, 시맨틱 위키 자체의 복잡도가 지나치게 증가하고, OWL 자원의 구조와 상관없는 자연어로 된 지식을 다루기 어려우며, 위키 페이지가 갱신되면서 링크가 사라질 경우, 사라진 링크와 연결된 페이지의 온톨로지에 접근할 수 없게 되는 단점이 있다.

따라서 이 논문에서는 위키의 일반 페이지에 온톨로지 메타 데이터가 포함되는 형태로 페이지의 내용을 구성하도록 설계했다. 이 구조는 일반 위키 텍스트의 내용과 온톨로지 메타 데이터 사이의 링크가 독립적으로 존재하는 단점이 있다. 그러나 위키 페이지의 내용과 온톨로지 메타 데이터가 동일한 물리적 위치에 함께 저장되며, 현실적으로 먼저 언급한 구조보다 유연하게 지식을 나타낼 수 있다는 장점이 있다. 이러한 구조에서 위키 페이지의 일반 텍스트 내용을 그 페이지가 가지고 있는 온톨로지에 대한 주석으로 생각할 수 있다. 그림 2는 시맨틱 위키의 편집 화면으로, 자연어와 위키 마크업 언어로 구성된 내용과 함께 {{{ }}}로 둘러싸인 영역에 OWL Concrete Abstract Syntax[11,12]로 표현된 온톨로지를 볼 수 있다.

시맨틱 탐색을 가능하게 하기 위하여 온톨로지 메타 데이터가 포함된 페이지의 경우, 사용자가 페이지를 조회할 때 페이지 하단에 페이지가 포함하는 온톨로지를 참조하는 다른 페이지의 자원에 대한 정보와 이동 가능한 링크를 표시하게 하였다. 이렇게 함으로써 사용자는 전통적으로 위키에서 링크로 알려진 CamelCaseLink 뿐만이 아닌, 페이지에 포함된 온톨로지가 참조하는 다른 문서를 조회할 수 있다. 이 링크를 이용하여 사용자는 레이블이 있는 양방향 링크를 따라서 위키 페이지를 탐색할 수 있다. 사용자가 이 링크를 따라서 탐색을 하면, 해당 페이지의 온톨로지 메타 데이터에 대한 정의와 함께 그와 관련된 자연어로 작성된 설명을 볼 수 있다. 따라서 사용자는 생소한 개념을 접하더라도 그 개념에 대하여 쉽게 정보를 얻을 수 있다.

위키의 가장 두드러진 특징은 누구나 임의의 페이지 내용을 편집할 수 있다는 것이다. 시맨틱 위키에서는 일반 페이지를 편집하는 창을 통하여 OWL Concrete Abstract Syntax로 온톨로지를 표현할 수 있다. 따라서 어떤 사용자라도 표현된 온톨로지의 내용을 일반 위키 페이지를 편집하는 것과 마찬가지로 별다른 특별한 프로그램 없이 웹에서 자유롭게 생성하고 수정할 수 있다. OWL Concrete Abstract Syntax에서 외부의 자원을 참조하는 방식은 OWL RDF/XML에서 사용하는 방

```

A '''planetary system''' consists of at least one star and various orbiting
objects (such as asteroids, comets, moons, and planets). Another name for this
collection is solar system, not to be confused '''Solar system''', the
planetary system that contains Earth and Sol.
{{{
  Ontology(
    Class(1:PlanetarySystem partial)
    Class(1:Star partial)

    DatatypeProperty(1:semiMajorAxis domain(1:Star))
    DatatypeProperty(1:siderealOrbitPeriod domain(1:Star))
    DatatypeProperty(1:avgOrbitalSpeed domain(1:Star))
    DatatypeProperty(1:surfaceTemp domain(1:Star))
    DatatypeProperty(1:surfacePressure domain(1:Star))
    ObjectProperty(1:belongsTo domain(1:Star) range(1:PlanetarySystem))
    ObjectProperty(1:hasSolarStar domain(1:Star))

    ObjectProperty(1:hasPlanet domain(1:PlanetarySystem))
    Class(1:Planet partial 1:Star)
    ObjectProperty(1:hasSatellite domain(1:Planet) range(1:Star))

  ) }}}
== Origin and evolution of planetary systems ==

Planetary systems are generally believed to form as part of the same process
    
```

그림 2 시맨틱 위키의 편집화면

식과 동일하다. 따라서 어떠한 온톨로지 데이터라도 RDF 또는 OWL RDF/XML 형태로 표현되어 있으면 시맨틱 위키에서 이를 자유롭게 참조할 수 있고, 이를 참조하는 온톨로지를 생성할 수 있다.

3.2 시맨틱 위키의 구조

시맨틱 위키는 그 자체로서 하나의 웹 사이트이다. 따라서 모든 작업은 웹 인터페이스를 통한 사용자의 요청에 의해 이루어진다. 시맨틱 웹 기반구조는 내용의 조회, 생성, 편집, 삭제, 검색과 온톨로지 메타데이터 출력의 모든 기능을 웹에서 처리한다. 이러한 기능을 수행하도록 본 논문에서 제안하는 시맨틱 위키 시스템의 구조는 그림 3과 같다.

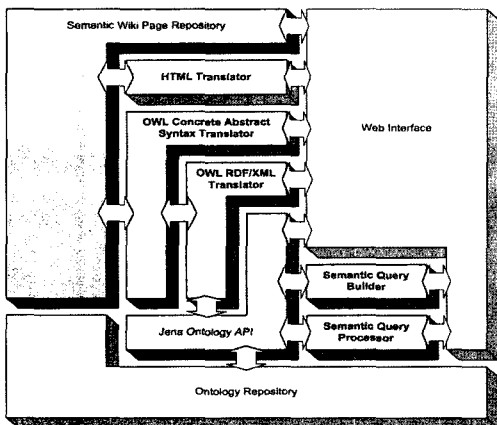


그림 3 시맨틱 위키 시스템의 구조

시맨틱 위키 페이지 저장소(Semantic Wiki Page Repository)는 일반 위키 문서에 OWL Concrete Abstract Syntax로 표현된 온톨로지가 포함된 형태의 시맨틱 위키 페이지를 저장하는 곳이다. 이렇게 저장된 문서를 웹 인터페이스(Web Interface)를 통하여 사용자에게 보여준다.

HTML 변환기(Translator)는 시맨틱 위키 페이지의 내용 중 일반 텍스트에 해당하는 부분을 HTML로 변환하고, 페이지가 포함하고 있는 온톨로지에 대한 정보를 HTML로 표시하여 웹 인터페이스에 전달한다.

OWL Concrete Abstract Syntax 변환기(Translator)는 시맨틱 위키 페이지에 포함된 온톨로지를 추출하는 모듈이다. 이렇게 추출된 온톨로지는 OWL RDF/XML 변환기(Translator)를 이용하여 RDF/XML 형태로 변환이 가능하다. 외부에서 OWL RDF/XML로 변환된 온톨로지를 요청할 때 OWL RDF/XML 변환기를 통하여 웹 환경을 통해 제공한다.

사용자에 의해 시맨틱 위키 페이지가 생성, 편집, 삭제될 때마다 해당 페이지에 포함된 온톨로지의 내용이 Jena 온톨로지 API[13]를 통하여 온톨로지 저장소(Ontology Repository)에 저장된다. 온톨로지 저장소에는 온톨로지의 내용이 (Subject, Predicate, Object) 형태의 트리플 단위로 저장된다. 이렇게 저장된 트리플은 시맨틱 조회와 검색을 할 때 이용된다.

사용자는 웹 인터페이스가 제공하는 시맨틱 질의 생성기(Semantic Query Builder) 기능을 이용하여 트리플 형태로 표현된 시맨틱 질의를 생성할 수 있으며, 이

렇게 생성된 질의는 시맨틱 질의 처리기(Semantic Query Processor)를 통해 수행되어 결과를 볼 수 있다.

사용자는 웹 인터페이스를 통하여 온톨로지와 자연어로 구성된 내용을 자유롭게 생성, 편집, 삭제할 수 있으며, 온톨로지의 내용을 OWL Concrete Abstract Syntax 또는 OWL RDF/XML, HTML의 형태로 조회할 수 있고, 시맨틱 질의에 대한 결과를 볼 수 있다.

#### 4. 시맨틱 위키의 구현

시맨틱 위키는 자바 언어를 사용하여 Apache Tomcat 어플리케이션 서버에 탑재되는 Servlet/JSP 형태로 작성되었다. 자바 언어를 사용하면 프로그램을 플랫폼에 독립적으로 작성할 수 있으며, 기존에 존재하는 Jena, Sesame, RDFAccess, OWLAP와 같은 시맨틱 웹을 프로그래밍할 때 유용하고 편리한 라이브러리를 사용할 수 있다.

##### 4.1 시맨틱 위키 페이지의 저장과 출력

시맨틱 위키는 일반 위키 페이지 내용뿐만이 아닌 시맨틱 온톨로지 메타 데이터도 함께 저장하고 불러올 수 있다는 점에서 기존의 위키와 다르다. 이 절에서는 3장에서 논의한 사항들이 구체적으로 어떻게 구현되었는지

설명한다.

##### 4.1.1 시맨틱 위키 페이지와 마크업 언어

시맨틱 위키에서 사용하는 위키 마크업 언어는 매우 단순하며, 일반적인 위키가 그러하듯 대부분 시각적 효과를 위한 태그이다. 표 1은 위키 마크업 언어에서 사용되는 태그를 나열하고 있다. 이러한 위키 마크업 언어를 이용한 위키 페이지의 내용은 그림 4와 같이 XML 형태로 저장된다.

위키 페이지의 루트 요소(root element)인 <page> 요소는 시맨틱 위키 페이지의 내용을 담고 있는 <content> 요소와 시맨틱 위키 페이지임을 나타내는 <des-

표 1 위키 페이지 마크업 언어

항목	태그 사용 예
강조	"emphasize"
굵은 글씨	"bold"
제목	== Title ==
목록	^^^* (^ = ', 0x20 공백문자)
수평구분	----
링크	CamelCaseLink 또는 http://hostname 직접입력
온톨로지 메타 데이터	{{{ Class(l:Mouse partial l:Animal) }}}}

```
<?xml version="1.0" encoding="UTF-8"?>
<page name="MusicGenres">
<content id="110055486295415470511" date="Tue Nov 16 06:41:02 KST 2004"
changedBy="166.104.219.80" hasOntology="true">
== Genres of the Popular Music ==
{{{
Namespace(am =
&lt;http://hammond.hanyang.ac.kr:8080/semwiki/owlxml/AboutMusic#&gt;)

Ontology(
)}}
* Jazz {{{ Class(l:Jazz partial am:PopularMusic) }}}
* Latin {{{ Class(l:Latin partial am:PopularMusic) }}}
* New Age {{{ Class(l:NewAge partial am:PopularMusic) }}}
* R&B {{{ Class(l:RnB partial am:PopularMusic) }}}
* Rap {{{ Class(l:Rap partial am:PopularMusic) }}}
* Rock {{{ Class(l:Rock partial am:PopularMusic) }}}
{{{ ) }}}
'''See Also''' AboutJazz
</content>
<islab:descriptor xmlns:islab="http://islab.hanyang.ac.kr/damls/descriptor"
creator="Semantic Wikiwiki" contact="hjkim@islab.hanyang.ac.kr"/>
</page>
```

그림 4 XML 형태로 저장된 시맨틱 위키 페이지

criptor> 요소로 구성된다. <content> 요소는 내용의 고유한 식별자인 id, 작성된 날짜와 시간을 알 수 있는 date, 작성한 사람에 대한 정보를 알 수 있는 changedBy, 그리고 온톨로지 메타 데이터가 있는지 여부를 알 수 있는 hasOntology 속성을 가지고 있다. 하나의 <page> 요소는 여러 개의 <content> 요소를 가질 수 있으며, 페이지가 갱신될 때마다 새로운 <content> 요소를 추가하기 때문에 <content> 요소의 비교로 해당 페이지가 갱신된 이력에 대하여 알 수 있다. <content> 요소의 id 속성은 고유한 식별자이며 동시에 해당 내용이 파일에 기록되는 순간의 타임 스탬프이다.

#### 4.1.2 시맨틱 위키 페이지의 생성과 갱신

기본적으로 위키 페이지의 생성과 갱신은 작업 이전의 페이지 존재 여부를 제외하면 동일한 작업이다. 원하는 내용을 편집한 후 저장하면 먼저 서버에 XML 형태로 페이지 데이터를 저장한다. 그리고 나서 해당 페이지가 온톨로지를 가지고 있을 경우, 온톨로지 저장소에서 해당하는 트리플을 갱신한다. 페이지를 생성할 때 기본적으로 rdf:, rdfs:, owl:, xsd: 와 같은 자주 쓰이는 주소 공간과 함께 해당 페이지의 기본 주소 공간을 !로 정해준다. 그리고 OWL Concrete Abstract Syntax의 문법에 따라 사용자가 자유롭게 추가적인 주소 공간을 지정할 수 있다.

OWL Concrete Abstract Syntax는 "OWL Web Ontology Language Semantics and Abstract Syntax[6]"에서 소개하는 OWL Abstract Syntax를 확장하고 Literal의 표현과 같은 부분을 보다 구체적으로 정의한 문법이다. 주소공간을 지정하기 위해 새로운 Namespace 라는 키워드가 추가되었으며 OWL Concrete Abstract Syntax에서는 N3와 다르게 prefix name에 '\_'를 허용하지 않는다. OWL Concrete Abstract Syntax는 기존의 OWL 데이터의 표현 방법이 XML이나 RDF로 되기 때문에 readability가 떨어져서 보다 간단하면서 개념을 명확히 표현하기 위한 구문구조이다. 이것은 W3C에서 표준으로 제안한 것이다. 하지만 기본적인 표현구조는 XML이므로 abstract syntax를 XML로 바꾸어주는 translator가 필요한데, 이 translator는 저자가 구현한 것이다.

#### 4.1.3 OWL Concrete Abstract Syntax와 OWL RDF/XML 온톨로지 조회

시맨틱 위키에서 <http://hostname/context/owlabs/PageName>으로 접근하면 위키 페이지 저장소에서 페이지의 내용을 읽어와서 {{{ }}} 위키 마크업 태그 사이에 존재하는 OWL Concrete Abstract Syntax의 내용을 취합하여 출력한다. 이 때 RDF, RDFS, OWL, XML

Schema와 해당 페이지에 대하여 주소 공간이 주어진다.

사용자 또는 에이전트가 <http://hostname/context/owlxml/PageName>로 접근하면 OWL RDF/XML로 작성된 온톨로지 메타 데이터를 얻을 수 있다. 이 XML 데이터는 바로 위에서 언급한 OWL Concrete Abstract Syntax를 RDF/XML 데이터로 변환한 것이다

#### 4.1.4 웹 브라우저를 통한 시맨틱 위키 페이지의 조회

웹 브라우저를 통하여 <http://hostname/context/PageName>으로 접근하면 그림 5와 같이 위키 페이지와 유사한 시맨틱 웹 위키 페이지를 HTML로 볼 수 있다. HTML로 변환된 일반 위키 페이지의 내용과 함께, 해당 페이지가 온톨로지를 포함한 경우 포함하고 있는 온톨로지 메타 데이터의 자원에 대한 정보를 페이지 하단에 출력한다. 정보를 출력하는 부분에서 굵은 글씨로 되어있고 !로 시작하는 것이 페이지가 포함하고 있는 자원이다.

온톨로지에 대한 정보는 다음과 같이 출력된다. 먼저 온톨로지 저장소를 검색하여 페이지에 포함된 자원이 저장된 트리플 중에서 Subject에 위치하는 경우에는 이 트리플에 대한 정보를 출력한다. 검색된 트리플의 Predicate이나 Object가 해당 페이지에 속하지 않는 경우, 즉 시맨틱 위키의 다른 페이지에 속하거나 시맨틱 위키 외부에 존재하는 자원인 경우, 직접 접근 가능한 HTML 링크를 표시한다. 해당 페이지에 속하지 않으나 시맨틱 위키의 다른 페이지에 속하는 경우에는 HTML로 조회할 수 있는 위키 페이지의 HTML 링크를 표시한다. 그런 다음, 포함하는 자원이 트리플에서 Object에 있는 경우에 해당하는 트리플을 검색하여 이에 대한 정보를 출력한다. 페이지에 포함된 모든 온톨로지 자원에 대하여 이러한 작업을 반복하여 정보를 출력한다. 이 정보를 이용하여 사용자는 해당 페이지에 포함된 온톨로지 메타 데이터 자원이 다른 자원과 어떠한 관계가 있는지 알 수 있으며, 시맨틱 위키에 포함된 자원의 경우 위키 페이지를 방문하여 보다 자세한 정보를 얻을 수 있다.

#### 4.2 시맨틱 검색

일반적인 위키 시스템의 경우, 페이지가 자연어로 작성된 일반 텍스트 형식의 정보만을 제공하기 때문에 위키에 대한 검색은 페이지 내용 또는 페이지 제목에 대한 단순한 텍스트 비교 또는 정규 표현식을 이용한 문자열 검색 방법이 사용된다. 시맨틱 위키는 이와 다르게 페이지에 온톨로지 메타 데이터 내용을 추가함으로써 단순한 텍스트 매칭이 아닌 의미 검색이 가능하다.

시맨틱 위키에서의 시맨틱 검색은 Jena 2 Ontology API를 이용하여 구현했다. OWL은 RDF의 관점에서 해

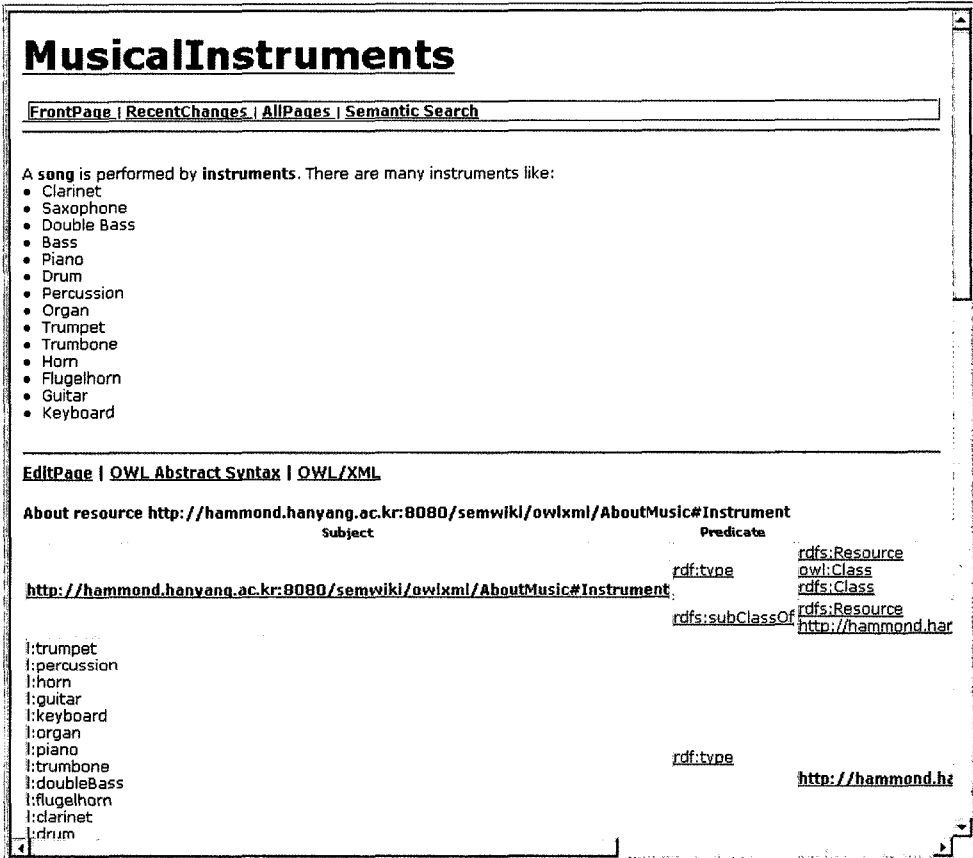


그림 5 시맨틱 위키 페이지 예제

석할 수 있으며, RDF는 온톨로지를 Subject, Predicate, Object로 구성되는 그래프 구조로 묘사한다.

시맨틱 위키에서 구현된 시맨틱 검색은 온톨로지를 그래프로 바라보고, 일종의 Depth-First 방식의 그래프 탐색 기법을 사용하여 사용자의 질의에 대한 검색을 수행한다. rdfs:subClassOf 뿐만이 아닌 owl:equivalentClass, owl:equivalentProperty, owl:sameAs와 같은 vocabulary에 대해서 확장된 탐색을 하며, 사이클이 생기는 경우에 대한 탐색은 하지 않는다. 시맨틱 위키는 Jena 2 API의 온톨로지를 저장소에 저장하고 불러오는 기능을 이용한다. 검색 조건은 (Subject, Predicate, Object)의 트리플 형태로 기술한다. 트리플에는 변수 또는 자원의 URI, Value 부분에는 Literal이 들어갈 수 있다.

그림 6에서 구현된 시맨틱 검색 인터페이스의 화면을 볼 수 있다. Subject, Predicate, Object에 해당하는 자원, 변수, 또는 값을 지정할 수 있으며, "Add a statement" 버튼을 누르면 지정된 값에 의한 Statement가 아래의 큰 텍스트 상자에 입력된다. 결과값을 보기를 원

하는 변수 한 개를 Result Variable에서 지정할 수 있으며 "Get Result" 버튼을 누르면 그림 7과 같은 결과를 볼 수 있다.

시맨틱 검색의 결과는 해당하는 자원의 URI와 함께 이 자원이 시맨틱 위키 내부의 자원일 경우, 자원이 포함된 위키 페이지에 접근할 수 있는 HTML 링크를 표시한다. 따라서 자원이 시맨틱 위키의 일부일 경우, 사용자는 XML로 표현된 정보뿐만이 아닌 위키 페이지 조회를 통한 보다 자세한 정보를 알 수 있다.

예를 들어 "Mick이 운전하는 차의 종류"를 묻는 질의는 그림 8과 같은 그래프 형태로 생각할 수 있다. 이 질의를 세분화하면 "Mick이라는 사람이 x를 운전하는데, x의 종류가 무엇인가?"라는 질의와 동일하다. 이것을 시맨틱 검색을 위한 트리플 구조로 변환하면 그림 9와 같으며, 결과값은 변수 ?y에 지정된다. 이 질의에 대한 검색 결과는 앞의 그림 6과 7에서 확인할 수 있다. 보다 복잡한 유형의 시맨틱 검색에 대한 예는 4.3.2절에 나와 있다.

### Query Statement Builder

**Subject**  
 Specific Resource:   
 Or Specific Variable:

**Predicate**  
 Specific Predicate:   
 Or Specific Variable:

**Object**  
 Specific Resource:   
 Or Specific Value:   
 Or Specific Variable:

**Result Variable**

---

**Query Statements**

```
<http://hammond.hanyang.ac.kr:8080/semwiki/owlxml/PeopleAndPets#Mick>,
<http://hammond.hanyang.ac.kr:8080/semwiki/owlxml/PeopleAndPets#drives>, ?x
```

그림 6 시맨틱 검색 인터페이스

Result	Wiki Page
<a href="http://ex/owlxml/Vehicle#van">http://ex/owlxml/Vehicle#van</a>	Vehicle
<a href="http://ex/owlxml/Vehicle#vehicle">http://ex/owlxml/Vehicle#vehicle</a>	Vehicle
<a href="http://ex/owlxml/Color#whiteThing">http://ex/owlxml/Color#whiteThing</a>	Color

그림 7 시맨틱 검색 결과

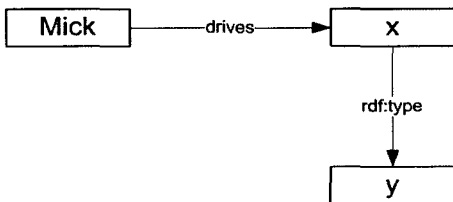


그림 8 "Mick이 운전하는 차의 종류"에 대한 질의의 그래프 표현

```

(<http://ex/owlxml/People#Mick>,
<http://ex/owlxml/Person#drives>, ?x),
(?x, <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>, ?y)
  
```

그림 9 "Mick이 운전하는 차의 종류"에 대한 질의의 트리플 표현

### 4.3 시맨틱 위키 응례

이 절에서는 구현된 시맨틱 위키 시스템과 예제 온톨로지[14]을 이용한 가상의 용례(Use Case)를 통하여 기존의 위키 시스템과 같은 지식 공동 저작 시스템과 어떤 차이점이 있는지 설명한다. 여기서 제시되는 각 온톨로지들은 각각 서로 다른 사용자들이 입력한 것이라고 가정한다. 즉, 각 온톨로지가 제시되는 순서대로 서로 다른 사용자에게 입력된다고 보면, 이전에 다른 사용자가 입력한 온톨로지를 바탕으로 새로운 사용자가 추가 온톨로지를 생성하여 입력할 수 있는 상황을 보여주는 것이다.

#### 4.3.1 온톨로지 공동 저작

식물을 좋아하는 Walt는 시맨틱 위키의 Plants라는 페이지에 식물에 대한 내용의 온톨로지를 기술하였다(그림 10).



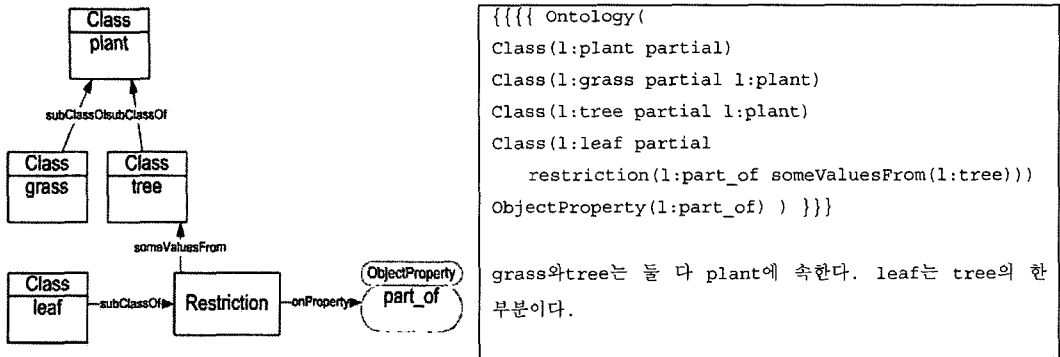


그림 10 Walt가 생성한 Plant 온톨로지와 위키 페이지

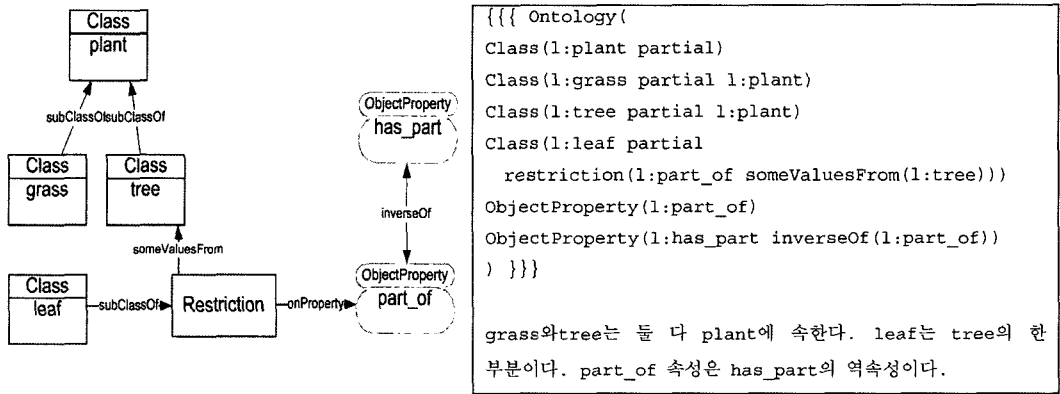


그림 11 Mick이 수정한 Plant 온톨로지와 위키 페이지

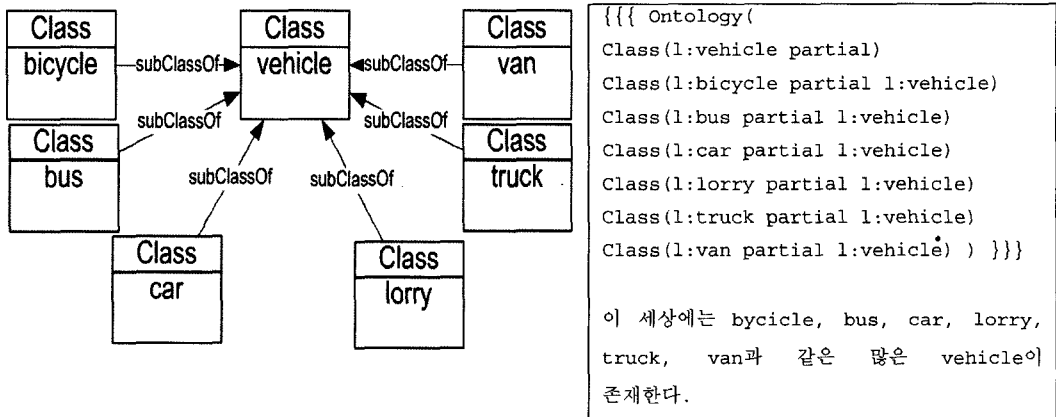


그림 12 Mick이 생성한 Vehicle 온톨로지와 위키 페이지

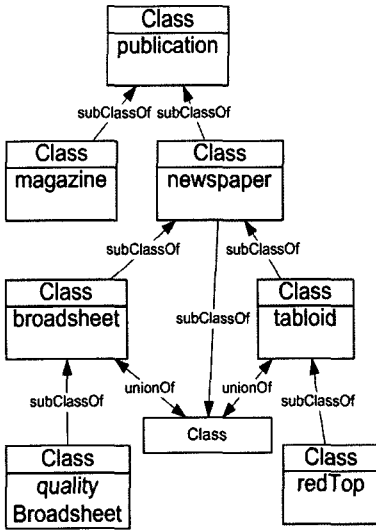
이것을 본 Mick은 part\_of 속성의 역속성(inverseOf)을 추가했다(그림 11). 그리고 운송수단에 대해 관심이 많아서 Vehicle 페이지를 생성하고 이에 대한 온톨로지와 자연어로 된 내용을 입력했다(그림 12).

신문에 관심이 많은 Fred는 Newspaper라는 위키 페이지를 생성하고 신문의 종류에 대한 온톨로지와 자연

어로 된 내용을 입력했다(그림 13).

동물을 좋아하는 Fred는 이번엔 Animals라는 위키 페이지를 생성하고 동물에 대한 온톨로지와 자연어로 된 내용을 입력했다(그림 14).

이번에는 Mick이 자기 자신에 대한 시맨틱 위키 페이지를 만들려다가 사람에 대한 온톨로지가 없다는 사



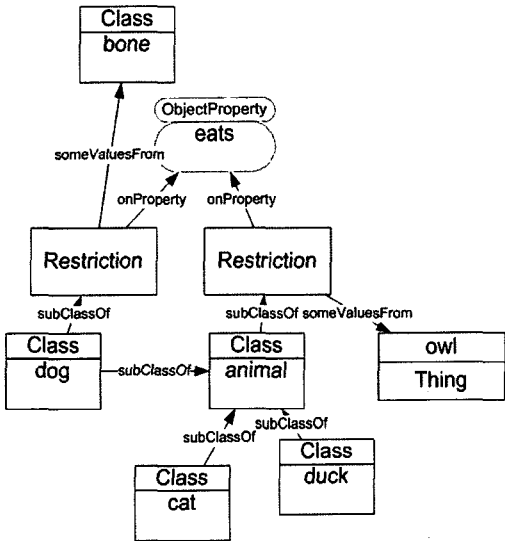
```

{{{ Ontology(
Class(1:publication partial)
Class(1:magazine partial 1:publication)
Class(1:broadsheet partial 1:newspaper)
Class(1:qualityBroadsheet partial 1:broadsheet)
Class(1:tabloid partial 1:newspaper)
Class(1:redTop partial 1:tabloid)
Class(1:newspaper partial
  unionOf(1:broadsheet 1:tabloid
    1:publication) ) }}}

```

Publication은 크게 magazine과 newspaper가 있다. newspaper는 다시 tabloid와 broadsheet으로 나뉘며, red top은 tabloid의 한 종류이고, quality broadsheet은 broadsheet의 한 종류이다. 즉, newspaper는 broadsheet과 tabloid를 포함한 것이다.

그림 13 Fred가 생성한 Newspaper 온톨로지와 위키 페이지



```

{{{ Ontology(
ObjectProperty(1:eats)
Class(1:animal partial
  restriction(1:eats
    someValuesFrom(owl:Thing)))
Class(1:duck partial 1:animal)
Class(1:cat partial 1:animal)
Class(1:bone partial)
Class(1:dog partial
  restriction(1:eats
    someValuesFrom(1:bone))) ) }}}

```

animal은 duck, cat, dog이 있으며, animal은 무엇인가를 먹는다. 예를 들어 dog은 bone을 먹는다.

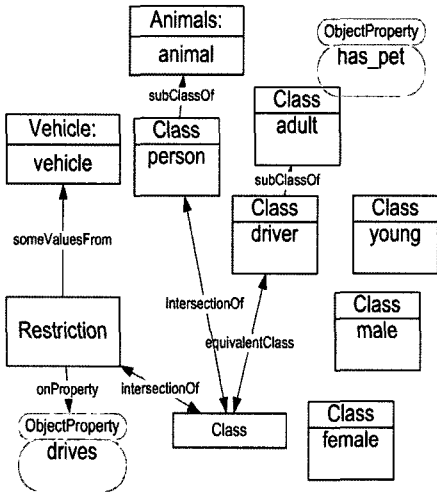
그림 14 Fred가 생성한 Animals 온톨로지와 위키 페이지

실을 깨닫고 Person 위키 페이지와 온톨로지를 생성한다(그림 15).

그 다음, Mick은 People 이라는 위키 페이지를 생성하여 Fred와 Walt와 자신에 대한 온톨로지 정보를 작성하였다(그림 16).

결과적으로 Mick, Fred, Walt가 시맨틱 위키에서 생성하고 편집한 페이지는 총 6개이며, 이 시맨틱 위키는 그림 17과 같은 온톨로지와 링크를 가지고 있다. 그림에

서 페이지 사이에 레이블이 붙은 링크는 다른 페이지의 온톨로지를 참조하는 시맨틱 웹 링크이며, 굵은 선으로 표시된 링크는 일반 위키 텍스트 부분에서 마크업 언어로 표시된 보통의 위키 링크이다. 이 가상의 사용 사례에서 단 3명의 사람이 공동으로 시맨틱 위키에 온톨로지와 자연어로 이루어진 내용을 작성한다고 해도, 사용하는 사람들이 신중하게 온톨로지를 잘 작성하면 매우 쉽게 복잡한 온톨로지를 생성할 수 있음을 알 수 있다.



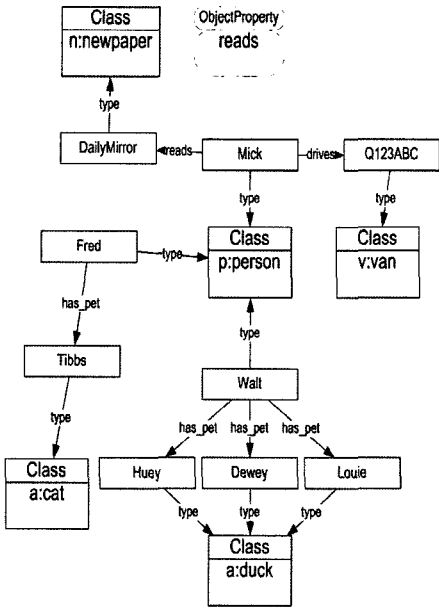
```

{{{ Ontology(
Class(l:person partial Animals:animal)
ObjectProperty(l:has_pet)
ObjectProperty(l:drives)
Class(l:male partial)
Class(l:female partial)
Class(l:young partial)
Class(l:adult partial)
Class(l:driver partial l:adult)
Class(l:driver complete
  intersectionOf(
restriction(l:drives
someValuesFrom(Vehicle:vehicle))
l:person)) ) }}}

```

person은 생물학적 분류상 animal이다. 사람은 male과 female, young과 adult로 구분할 수 있다. 보통 adult가 되어야 driver가 될 수 있으며 driver또한 person이고, driver는 [[Vehicle]]을 운전한다. 그리고 애완동물 [[Animals]]을 가질 수 있다.

그림 15 Mick이 생성한 Person 온톨로지와 위키 페이지



```

{{{
Namespace(p = <http://semwiki/owlxml/Person#>)
Namespace(a = <http://semwiki/owlxml/Animals#>)
Namespace(v = <http://semwiki/owlxml/Vehicle#>)
Namespace(n = <http://semwiki/owlxml/Newspaper#>)
Ontology(
  Individual(l:Dewey type(a:duck))
  Individual(l:Huey type(a:duck))
  Individual(l:Louie type(a:duck))
  Individual(l:Q123ABC type(v:van))
  Individual(l:Tibbs type(a:cat))
  Individual(l:Fred type(p:person)
    value(p:has_pet l:Tibbs))

  Individual(l:DailyMirror type(n:newspaper))
  Individual(l:Mick type(p:person)
    value(l:reads l:DailyMirror)
    value(l:drives l:Q123ABC))
  Individual(l:Walt type(p:person)
    value(l:has_pet l:Huey)
    value(l:has_pet l:Louie)
    value(l:has_pet l:Dewey))
)
}}}
```

그림 16 Mick이 생성한 People 온톨로지와 위키 페이지

### 4.3.2 시맨틱 검색

어느 날, 시맨틱 위키에 Minnie라는 할머니가 등장하여 그림 18과 같은 내용의 온톨로지를 OldLady 페이지에 작성하였다.

Minnie는 People 페이지에 그림 19와 같은 자신에 대한 온톨로지를 추가하였다.

시맨틱 위키를 탐색하던 Mick은 Minnie라는 사람에 대한 새로운 온톨로지가 추가되었다는 사실을 발견했다. 하지만 Minnie의 애완동물인 Tom의 type이 Thing으로 되어 있어서 구체적으로 어떠한 동물인지 알 수 없었다. 그래서 Mick은 시맨틱 검색 기능을 이용하여 그림 20과 같은 시맨틱 질의를 입력하여 이를 알아보기로

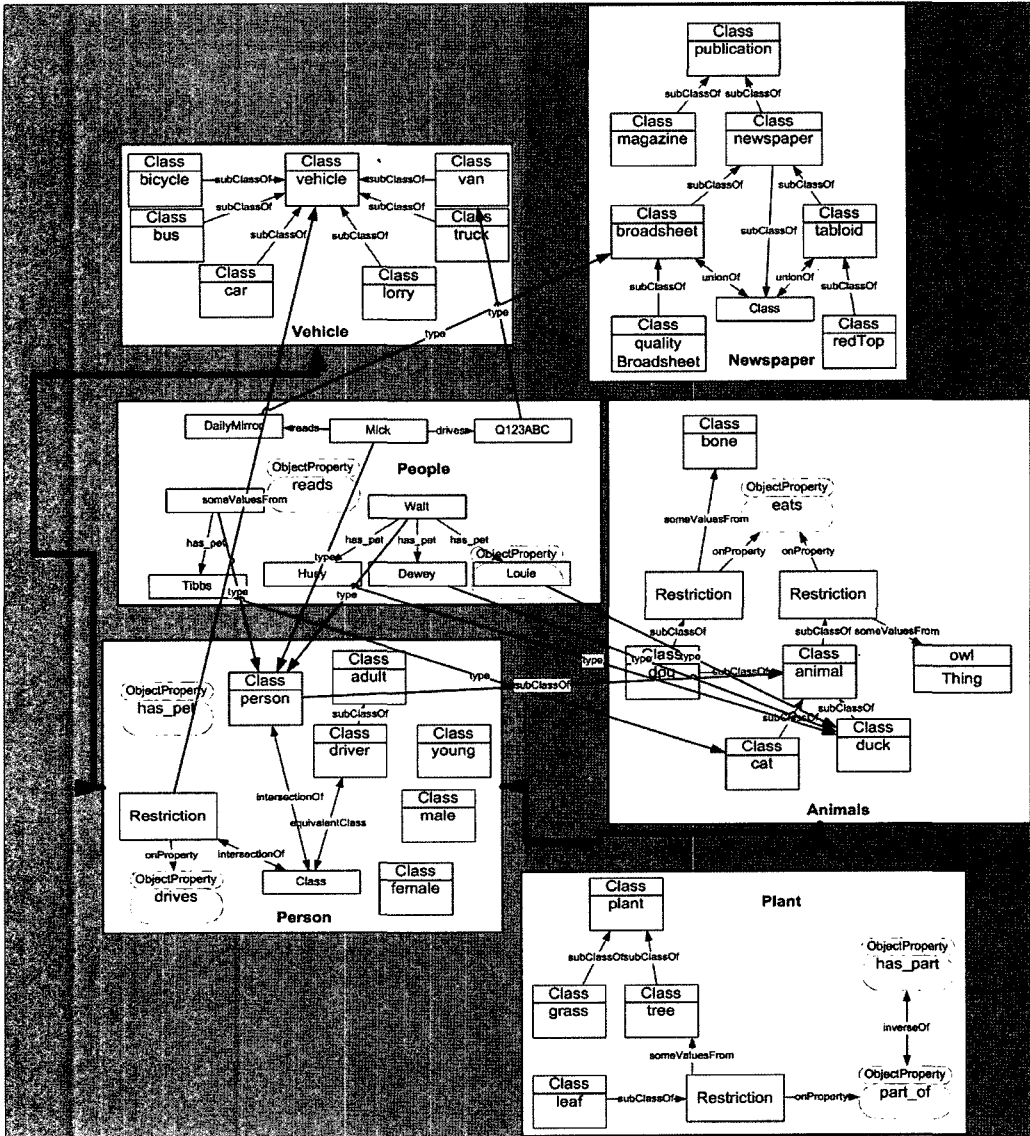


그림 17 전체 온톨로지와 위키 페이지

하였다. 이 질의에 대한 결과는 그림 21과 같다.

그림 19를 보면 Minnie의 애완동물인 Tom은 종류가 owl:Thing으로 되어있다. 이것은 Minnie가 직접 추가한 것이며, 이 정보만 가지고는 Tom이 구체적으로 어떤 동물인지 알 수 없다. 그림 18을 보면 oldLady는 elderly, female, person의 교집합을 이루는 class와 동등한 클래스임을 알 수 있다. 그런데 그림 19에 정의된 Minnie가 바로 elderly, female, person 모두에 해당하는 Individual이다. 그러므로 Minnie는 oldLady인 것이다. 그림 18을 다시 살펴보면 oldLady는 애완동물을 가지는 클래스의 하위 클래스인데, 애완동물을 여러 마

리를 가질 때, 동물의 종류가 적어도 하나는 animal인 클래스와, 모든 애완동물의 종류가 cat인 클래스의 교집합 클래스이다. 이 조건으로부터 oldLady의 애완동물은 animal이자 동시에 cat임을 알 수 있다. 따라서 시맨틱 검색을 통한 결과는 원래 정의된 owl:Thing과 더불어 OldLady 온톨로지서 정의된 oldLady의 조건에 따라서 cat과 animal도 검색 결과로 출력된 것이다. 이러한 검색 결과는 일반 텍스트 데이터에 대하여 수행하는 텍스트 매칭 기법으로는 보여줄 수 없다.

### 4.3.3 웹에서 지식 교환과 정보 공유

OWL의 특성상 인터넷에 URI로 존재하는 모든 온톨

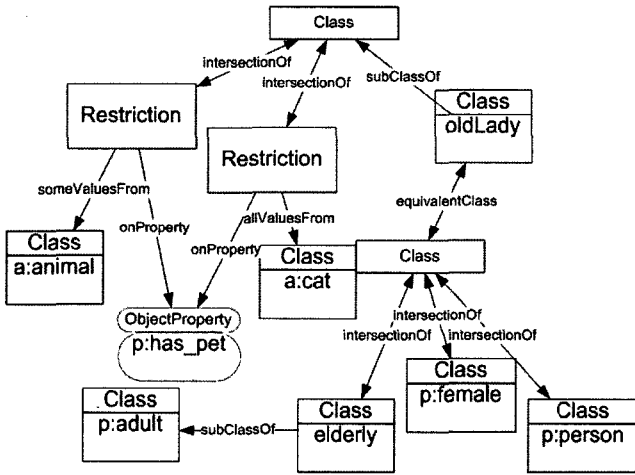


그림 18 Minnie가 작성한 OldLady 온톨로지

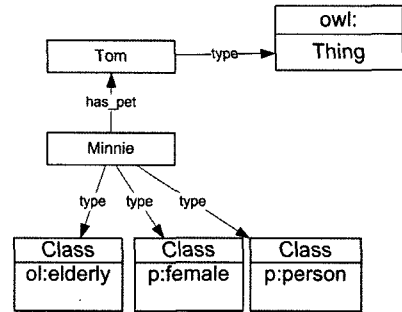


그림 19 Minnie가 People에 추가한 온톨로지

```

Query Statements:
(<http://semwiki/owlxml/People#Minnie>, <http://semwiki/owlxml/Person#has_pet>, ?x),
(?x, <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>, ?y)
Result Variable:
?y
    
```

그림 20 Minnie의 애완동물 종류 검색을 위한 시맨틱 질의

```

http://www.w3.org/2002/07/owl#Thing
http://semwiki/owlxml/Animals#animal, http://semwiki/Animal
http://semwiki/owlxml/Animals#cat, http://semwiki/Animal
    
```

그림 21 Minnie의 애완동물 종류에 대한 검색 결과

로지 자원을 특별한 조작 없이 그대로 사용할 수 있다. OWL을 사용하는 시맨틱 위키 또한 다른 시맨틱 위키 또는 다른 정적인 OWL RDF/XML 문서의 온톨로지를 참조하는 것은 특별한 추가적인 장치 없이 가능하다. 페이지를 생성 또는 수정할 때 위키 내부의 자원에 대한 URI를 적는 것과 위키 외부의 자원에 대한 URI를 적는 것은 차이가 없다.

### 5. 결론

이 논문에서 제안한 시맨틱 위키는 기존 위키 시스템에 시맨틱 웹을 접목하여 공동체 단위에서 유용하게 사용할 수 있고 공통으로 사용하는 어휘와 정보, 온톨로지와 일반적인 지식을 공동으로 저작하고 공유할 수 있는 시스템이다.

시스템을 구현하기 위하여 자바 언어를 사용했으며, 일반 위키 페이지의 내용에 OWL 온톨로지가 포함되는 형태로 페이지 데이터를 관리하도록 하였다. 일반 위키

의 특징을 그대로 수용하면서 동시에 일반 페이지 링크와 시맨틱 웹 링크가 공존하는 형태가 되었고, 일반 위키 페이지의 내용은 포함된 온톨로지 데이터의 주석문과 같은 역할을 하게 되었다. 또한 온톨로지 정보를 이용한 시맨틱 검색으로 인하여 기존의 위키 또는 지식관리 시스템에서 불가능했던 의미 정보에 대한 검색이 가능해졌다.

그러나 페이지에 포함된 온톨로지와 위키 텍스트가 독립적으로 존재하는 문제점이 있다. 이는 시맨틱 정보를 기술할 수 있으면서 동시에 위키의 링크를 표시할 수 있는 새로운 마크 업언어를 고안하여 해결 가능하다. 그리고 인공지능 학습 기법을 이용하여 위키 페이지에 대한 주제를 자동으로 부여하고 분류하면 시맨틱 위키의 내용을 더욱 편하게 탐색할 수 있을 것이며, 시맨틱 위키에 추론 엔진을 접목하여 시맨틱 위키에 포함된 온톨로지에 표현된 의미 정보를 기초로 하여 보다 많은 정보를 유추하면 더욱 다양한 시맨틱 검색 기능이 가능

할 것이다.

OWL의 특성에 의하여 시맨틱 위키에서 외부에 존재하는 OWL 자원을 참조하는 것은 별도의 추가적인 장치 없이 가능하다. 그러나 외부에 어떠한 자원이 있는지 쉽게 알 수 있는 방법이 없다. 따라서 시맨틱 위키가 가지고 있는 자원에 대한 정보를 얻을 수 있는 표준화된 Protocol 또는 API, 시맨틱 위키가 가지고 있는 온톨로지와 함께 웹에 존재하는 다른 온톨로지 자원에 대한 정보를 제공할 수 있는 온톨로지 검색 엔진과 같은 시스템이 필요할 것이다.

시맨틱 위키에 수록되는 온톨로지는 좋은 방향으로 관리가 잘 되면 손쉽게 훌륭한 내용으로 가득 찰 수 있으나, 관리가 조금만 소홀해도 내용의 무결성에 위배되는 경우가 발생할 수 있다. 온톨로지 구축시에 온톨로지 엔지니어의 입장에서는 자신이 구축한 온톨로지에 대한 validation 과정을 반드시 거쳐야 한다. 이러한 온톨로지 validation 이슈는 사실 시맨틱웹이 일반에게 알려진 초기에는 별로 관심을 갖지 못한 부분이었지만 이제 어느 정도 온톨로지의 효용성이 나타나고 필요성이 인식된 시점에서는 온톨로지의 구축 분량 못지 않게 integrity도 중요한 부분으로 여겨지고 있다. 따라서 integrity에 대한 해결방법에 대한 시도는 온톨로지 추론도구의 상용화 제품이 출시된 근래에 와서 이루어지고 있으며 RacerPro나 Fact와 같은 온톨로지 추론 엔진을 이용해서 구축된 온톨로지 사이의 결함을 발견하고 수정하는 작업으로 이루어지고 있다. 현재까지 온톨로지 validation은 온톨로지 구축과 이용이 분리된 환경에서 구축을 담당하는 엔지니어에 의해 적용되었는데, 위키 시스템의 경우는 위키 데이터의 구축과 이용이 별개로 이루어지는 것이 아니고 통합된 형태로 이루어지기 때문에 validation 작업이 어려운 것이 사실이다. 시맨틱 위키가 다루는 온톨로지가 특정 도메인에 국한된 것이 아닌 일반적인 것이기 때문에 온톨로지 validation이 이루어지면 모든 도메인 정보에 대한 restriction 모듈을 구성해야 하고 현실적으로 이를 구성하는 것이 어렵다. 따라서 현재는 사용자의 적극적 참여를 유도하거나, 또는 소수의 전문가 집단에게 온톨로지 내용의 평가를 전담하게 하여 관리하도록 하는 방법이 현실성이 있을 것이라고 생각된다. 입력된 온톨로지의 내용이 논리적으로 맞는다 하더라도 잘못된 근거를 기반으로 수립된 논리적인 사항은 유용한 온톨로지라고 할 수 없다. 시맨틱 위키의 경우 입력된 데이터가 기계가 처리하기 용이한 메타 데이터의 형식을 취하고 있지만 한계가 있을 수 밖에 없는 것이다. 따라서 이러한 부분에 대해서는 기계가 스스로 판단을 내릴 수 없으므로 관리 인력의 개입이 필수적이다.

시맨틱 위키는 일반 사용자들을 시맨틱 웹의 세계로 안내하는 중요한 역할을 할 것이라고 확신한다. 그리고 P2P, 블로그, 위키, 게시판과 같은 시스템에 시맨틱 웹 요소가 결합되면 현재 웹이 시맨틱 웹으로 진화하는 과정을 더욱 촉진시킬 수 있을 것이다.

## 참고 문헌

- [1] Berners-Lee, T., Hendler, J. and Lassila, O., "The Semantic Web," *Scientific American*, 2001.
- [2] 최중민, "시맨틱 웹의 개요와 연구동향", 정보과학회지, 제21권, 제3호, pp. 4-10, 2003.
- [3] Maedche, A., *Ontology Learning for the Semantic Web*, pp. 11-27, Kluwer Academic Publisher, 2002.
- [4] Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M. and Horrocks, I., "The Semantic Web: the roles of XML and RDF," *IEEE Internet Computing*, Vol. 4, No. 5, pp.63-73, 2000.
- [5] McGuinness, D. L., and Harmelen, F., van, *OWL Web Ontology Language Overview*, World Wide Web Consortium(W3C) Recommendation, 2004, <http://www.w3.org/TR/2004/REC-owl-features-20040210/>.
- [6] Wikimedia Foundation, *Wikipedia: The Free Encyclopedia*, <http://wikipedia.org/>
- [7] 오우택, "지식 검색과 공유를 위한 온톨로지 기반 지식 관리 시스템 구축", 한양대학교 대학원, 2004.
- [8] Cunningham, W., *Portland Pattern Repository*, <http://www.c2.com/cgi/wiki>
- [9] Tazzoli, R., Castagna, P., and Campanini, S. E., "Towards a Semantic Wiki Wiki Web," *Proceedings of 3rd International Semantic Web Conference (ISWC2004)*, Hiroshima, Japan, 2004.
- [10] Reynolds, D., Cayzer, S., Shabajee, P., and Steer, D., "SWAD-Europe deliverable 12.1.8: SWAD-E Demonstrators - Lesson Learnt," *World Wide Web Consortium*, 2004, <http://www.w3.org/2001/sw/Europe/reports/demo-lessons-report>
- [11] Patel-Schneider, P. F., Hayes, P., and Horrocks, I., *OWL Web Ontology Language Semantics and Abstract Syntax*, World Wide Web Consortium (W3C) Recommendation, 2004, <http://www.w3.org/TR/owl-semantics/>
- [12] Bechhofer, S., Patel-Schneider, P. F., and Turi, D., *OWL Web Ontology Language Concrete Abstract Syntax*, University of Manchester, 2003, <http://owl.man.ac.uk/2003/concrete/latest/>
- [13] HP Laboratory, *Jena - A Semantic Web Framework for Java*, 2004, <http://jena.sourceforge.net/>
- [14] I. Horrocks, "Example Ontology," *Tutorial of 2nd International Semantic Web Conference (ISWC2003)*, 2003, <http://www.cs.man.ac.uk/~horrocks/ISWC2003/Tutorial/people+pets.abs>



김 현 주

2003년 한양대학교 전자컴퓨터공학부 졸업(학사). 2005년 한양대학교 대학원 컴퓨터공학과 졸업(석사). 2005년~현재 삼성 SDS 전자CPC파트. 관심분야는 시맨틱 웹, 정보검색, 인공지능, SOA



최 중 민

1984년 서울대학교 컴퓨터공학과 졸업(학사). 1986년 서울대학교 대학원 컴퓨터공학과 졸업(석사). 1993년 State University of New York at Buffalo, Computer Science 졸업(박사). 1993년~1995년 한국전자통신연구원(ETRI) 인공지능연구실 선임연구원. 1995년~현재 한양대학교 컴퓨터공학과 교수. 관심분야는 지능형 에이전트, 시맨틱 웹, 인공지능, 웹정보추출