

시소러스를 기반으로 한 온톨로지 시스템 구현에 관한 연구

A Study on the Thesaurus-based Ontology System for the Semantic Web

정도현(Do-Heon Jeong)*, 김태수(Tae-soo Kim)**

초 록

본 연구는 W3C를 중심으로 시맨틱 웹을 위한 표준 온톨로지 언어로 제안된 OWL(Web Ontology Language)을 바탕으로 패싯형 시소러스인 AAT(Art and Architecture Thesaurus)의 개념 체계를 이용한 온톨로지 스키마를 구현하여 시맨틱 웹 환경의 온톨로지 기반 시스템을 구축하였다. 특히, 개념의 범위를 AAT의 Furniture 패싯 이하의 용어에 한정하였고 이 중 개념관계가 다양하고 많은 어휘를 가진 하위 패싯인 Chair 개념을 중심으로 시스템 구현 및 테스트를 하였다. 이 시스템은 온톨로지로 정의된 개념을 검색하며, 동의어에 대해 항상 우선어(Preferred term)를 제시하여 어휘를 제어한다. 또한 질의어를 중심으로 한 용어간의 관계를 이용자에게 제시하고 관련어 및 개념의 분류기준 속성을 제시한다. 그 밖에 개념에 인스턴스 값을 입력하고 이를 RDF 문서로 재생산하는 시스템을 구축하여 온톨로지 시스템 응용의 일례를 제시하였다. 이 과정에서 여러 온톨로지를 도입하였고, 인스턴스 값을 담기위한 컨테이너 기반으로 메타데이터 요소를 사용하였다.

ABSTRACT

The purpose of the study was to construct a system based on the semantic web environment's ontology by utilizing the ontology schema derived from the facet-type Art and Architecture Thesaurus(AAT). The aforementioned ontology schema is based on the Web Ontology Language(OWL), which is being widely considered the standard ontology language for the W3C-centered semantic web environment. Also, the concepts were limited to terms within AAT's Furniture Facet, and the system was tested using the Chair concept, which is a lower-level facet that has a diverse conceptual relationship and broad vocabulary base. The ontology system is capable of searching for concepts, while controlling the search results by always providing a 'Preferred term' for synonymous terms. In addition, the system provides the user with first, a relationship between the terms centered around the inquiry, and second, related terms along with their classification properties. Also, the system is presented as an application example of the ontology system that constructs a information system that intakes an Instance value and reproduces it into a RDF file. During this process, utilization of multiple ontologies was introduced, and the stored Instance value's meta-data elements were used.

* 연세대학교 대학원 문헌정보학과(heon@lis.yonsei.ac.kr)

** 연세대학교 문헌정보학과(btrees@yonsei.ac.kr)

■ 논문 접수일 : 2003. 8. 21

■ 게재 확정일 : 2003. 9. 4

키워드 : 시맨틱 웹, 온톨로지, RDF, RDF 스키마, XML 스키마, 메타데이터, semantic web, Ontology, meta-data

1 서 론

지식을 축적하고 활용하기 위한 지속적인 노력을 통하여, 시맨틱 웹(Semantic Web)이라는 개념에까지 이르게 되었는데, 이 개념은 인간의 지식을 정형화하기 위한 정보조직의 욕구에서 비롯된 산물로써, 언어로 표현된 개념과 지식을 정형화하는 방법을 제시하고 이를 표준 웹 환경으로 통합하여, 미래의 웹 환경을 일종의 통합된 목록시스템으로 구현하려는 노력이라고 할 수 있다. 최근 이러한 연구는 월드와이드웹(World Wide Web)의 탄생에 공헌한 팀 버너스리(Tim Berners-Lee)를 중심으로 한 웹 컨소시엄(www.w3c.org)과 학계 및 업계에서 활발히 전개되고 있다.

본 연구는 시맨틱 웹을 구현하기 위한 도구인 온톨로지 언어를 이용하여 기존의 시소러스 어휘간의 관계를 온톨로지 체계로 전환하였고, 이를 기반으로 한 응용시스템을 제안하는 것을 목적으로 한다. 이를 위해 기존의 시소러스 중 비교적 개념 정의가 체계화된 패싯형 시소러스인 Art and Architecture Thesaurus(AAT)를 선정하고, 구현을 위한 온톨로지 언어의 표준안을 분석하여 실제 온톨로지 문서를 작성하는 과정이 필요하였다. 주요 개념 관계를 구현하기 위해 AAT의 7개 패싯 중 하나인 Object 패싯의 Furniture 패싯으로 한정하였으며, 특히 이중 어휘간 개념관계가 다양하게 나타나는 Chair 패싯을 중심으로 구현하였다.

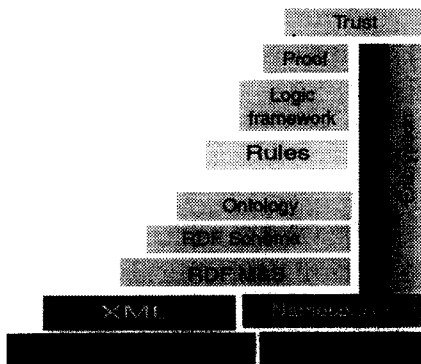
용어간의 개념관계에 있어 계층관계 표현과 동의어, 상하위어, 관련어 처리 등에 중점을 두었으며, 기타 가능한 속성표현을 제시하였다. 개념에 대한 인스턴스 값을 부여하기 위해 더블린 코어와 VRA 핵심 범주 등의 메타데이터를 이용하고 복수 온톨로지를 도입하여 각각의 온톨로지를 독립적인 제휴방식으로 운용하였다.

단, axiom을 포함한 온톨로지 언어의 많은 부분이 향후 발전될 추론시스템에 많은 부분을 의존하고 있으므로, 현재 온톨로지 시스템의 완전한 성능을 보여주기 힘든 한계가 있었다. 본 연구는 실제 구현 및 응용연구가 부족했던 분야에서의 선행연구로써 의미를 가지며, 온톨로지 언어로 시스템을 구축하기 위한 기존의 많은 선행연구들과 다른 방식인 온톨로지를 이용한 인스턴스 입력 시스템의 구현 사례를 제시했다는 점이 연구의 의의라 할 수 있을 것이다.

2 이론적 배경

시맨틱 웹을 실현하기 위한 방법으로 현재 제안된 가장 일반적인 연구 방향은 <그림 1>에서 보는 바와 같이, 먼저 하위 수준의 개념을 정의하고 이를 기반으로 다음 수준의 개념을 정의하는 계층구조로 설정되어 있다. 하위 수준에서 가장 중요한 역할을 하는 것으로는 메타데이터와 RDF(S) 등이 있으며, 온톨로지 기술은 이러한 연구에 기반하여 구현되도록 제안

되고 있다. <그림 1>은 최하위 수준에 자원의 주소를 지정하기 위한 URI가 있고 이를 기반으로 XML과 이름공간(Name-space), RDF(S), 온톨로지의 순서로 계층이 구성되어 있음을 보여주고 있다(Berners-Lee, Hendler, and Lassila 2001). 상위계층인 Logic에 대해서는 인공지능 분야의 추론 연구 등 일부 연구가 진행 중에 있으며, 시맨틱 웹의 보안과 신뢰성에 대한 Proof와 Trust 분야는 개념 정도만 언급되고 있다. 본 연구에서는 현재 활발히 연구가 진행되고 있는 시맨틱 웹의 온톨로지 단계까지의 내용이 주로 논의될 것이다.



<그림 1> 팀 버너스-리가 제안한 시맨틱 웹의 구조

2.1 메타데이터와 RDF(S)

시맨틱 웹에서 중요한 역할을 담당하는 것은 메타데이터로써, 현재 더블린 코어가 대표적이다. 메타데이터가 웹 자원에 대해 기계 가독형 정보를 부여함으로써 이를 바탕으로 수준 높은 검색기능이 가능하다. 그런데 이를 위해서는 메타데이

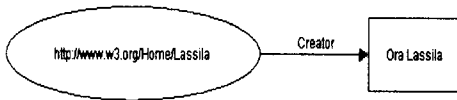
터를 기술하고 교환하기 위한 도구가 필요하게 되었고, 더블린 코어를 비롯한 다양한 메타데이터를 표현할 수 있는 RDF(Resource Description Framework)가 출현하게 되었다. RDF는 XML의 문법적인 기반 위에 탄생했지만 XML은 문서의 구조와 형식을 정의하는 반면, 문서에 포함된 데이터의 내용은 해석하지 않기 때문에 메타데이터를 상호운용하기 위해서는 결국 RDF의 역할이 중요하다(Hjelm 2001). 즉 RDF는 메타데이터를 처리하기 위한 기초이며 웹에서 기계가 이해할 수 있는 정보를 교환하는 응용프로그램들 간에 상호운용성을 제공하는 추상구문으로써, 인간이 이해할 수 있고(human-readable), 기계가 처리할 수 있는(machine-processable) 어휘를 정형화함으로써 메타데이터의 어휘를 다시 사용하거나 교환할 수 있도록 하는 것이 RDF(S)의 핵심이다(김태수 1998). RDF 데이터모델은 웹상의 정보자원을 기술하고 이들 간의 관계를 명시하는 모델이며, RDF 스키마는 RDF를 사용하고 RDF 어휘를 기술하는 방법을 정의하는 개념명세를 제공하고 있다.

RDF는 세 가지 요소로 이뤄진 구조를 기본으로 하여 지식을 표현한다. '개체(object)-속성(attribute)-값(value)'의 구조를 가지며, A(O,V)로 표현할 수 있다. 또한 이 구조는 '자원(resources)-속성(property types)-문 또는 값(statements)'의 구조 혹은 '주어-서술어-목적어'의 구조로 파악

되기도 한다. 즉 RDF의 3-튜플 모형은 온톨로지 기반의 시맨틱 웹을 구축하기 위한 기초 구문형식이다(Berners-Lee, Hendler, and Lassila 2001).

다음은 “Ora Lassila는 자원 <http://www.w3.org/Home/Lassila>의 작성자이다”라는 문장을 구성요소로 분석하여 표현한 예이다(Lassila, and Swick 1999).

주어(자원)	http://www.w3.org/Home/Lassila
서술어(속성)	Creator
목적어(문, 값)	Ora Lassila



〈그림 2〉 3-튜플 구조의 RDF 기본 모델

2.2 온톨로지 언어

RDF(S)는 웹 자원을 표현하기 위해 어휘, 구조 및 제약조건에 대한 정의 방법만을 제공하기 때문에 에이전트나 관련 응용 프로그램에 의해 지식을 처리하기가 어렵다. 따라서 의미를 담고 있는 표현구문인 axiom을 사용함으로써 지식을 표현하고 절차적 추론과정을 수행할 수 있는 기반인 온톨로지 언어(Ontology Language)가 등장하게 되었다. 현재로는 XML 기반의 XOL(Ontology Exchange Language), OML(Ontology Markup Language), SHOE

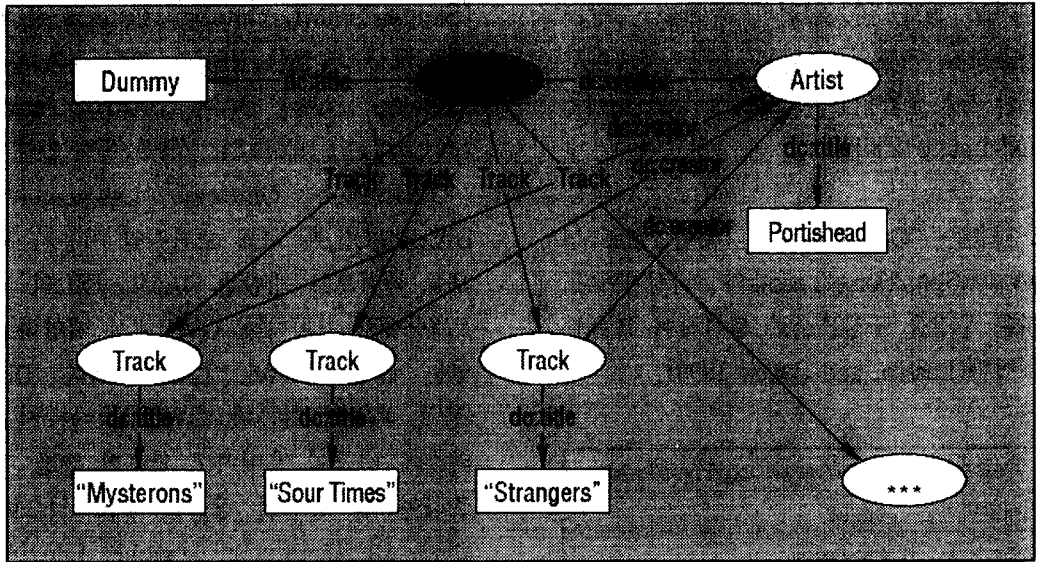
(Simple HTML Ontology Extensions)와 W3C에서 제정한 RDF와 OIL(Ontology Inference Layer), DAML(DARPA Agent Markup Language)+OIL, OWL 등이 있다.

OWL(Web Ontology Language)은 DAML+OIL에 비해 보편적인 이용자 확보를 목적으로 개발된 DAML+OIL의 새로운 개정 언어이다. OWL에서 제안하고 있는 유형은 OWL Lite와 OWL DL, OWL Full이 있는데, OWL Lite는 시소러스에 접근이 용이하고 단순한 면을 강조하여 웹 응용에 강점을 갖고 있으며, OWL DL(Description Logic)은 Lite보다 좀더 논리적인 표현을 위한 온톨로지이다. 마지막으로 OWL Full은 표현력에 있어서 가장 풍부하며 RDF의 구문을 모두 허용하고 있다. OWL Lite는 웹 온톨로지 언어로써 이용하기에 용이하며, 특히 시소러스의 개념에 접근하기 쉽다는 장점이 있다(Smith, Welty, and McGuinness 2003).

2.3 선행 연구

2.3.1 음악분야의 시스템: MusicBrainz

MusicBrainz는 2003년 2월 현재 3만여 명의 음악가와 5만여 개의 앨범, 65만개 이상의 트랙정보를 가지고 있는 음악분야의 메타데이터 시스템이다(Swartz 2003). 각 회사 혹은 각 개인마다 각자의 방식대로 작성한 음악에 대한 메타데이터 정보들을 간단한 온톨로지 표준화를 통해 체계적으로 관리한다. 음악가, 앨범, 트랙에



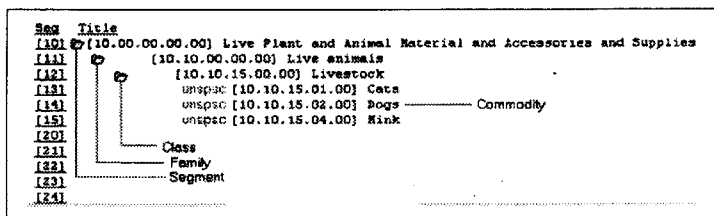
〈그림 3〉 Portishead 앨범에 대한 MusicBrainz RDF 데이터 구조

대해 URI를 부여하고 정보를 RDF로 기술한다. URI로 객체들이 유일하게 결정되므로, 메타데이터의 중복을 피할 수 있고 오류의 가능성을 줄일 수 있다. 또한, 외부의 응용애플리케이션이 RDF로 기술된 정보를 해석하고 이를 이용할 수도 있다. 이 시스템에서 온톨로지 언어는 메타데이터를 통합하는 역할을 한다. 분산된 메타데이터를 매핑하는 방법으로 온톨로지 언어를 상위에 두고 구문적으로 상호매핑을

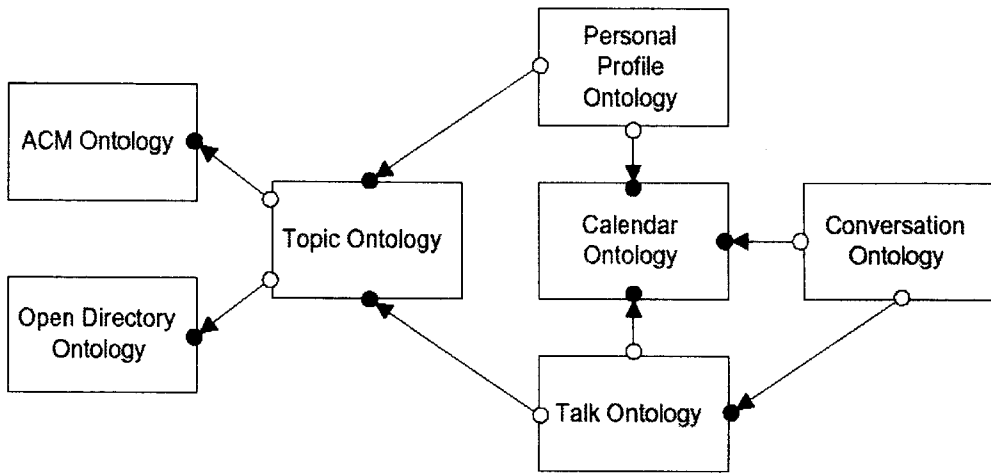
처리하고 있다.

2.3.2 상품분류체계 시스템: UNSPSC

생산자와 소비자의 요구를 최대한 대응시키기 위해, 전자카탈로그와 같이 여러 판매 사이트간 제품 정보를 통합하여 공통으로 분류하는 체계로써 UNSPSC를 들 수 있다. 이는 전자상거래 및 비즈니스에 참여하는 모든 물품과 서비스를 계층적으로 분류하는 표준이다. 제품분류를 위해



〈그림 4〉 UNSPSC의 제품분류 체계(<http://eccma.org/unspsc>)



<그림 5> ITTALKS 시스템의 다양한 온톨로지 간의 상호관계

온톨로지를 이용하며, 제품에 대한 공통의 어휘와 계층적 구조로 표준화한다. 이 온톨로지를 이용하여 여러 사이트들이 제품을 동일하게 지칭할 수 있으며, 공통의 어휘로 정보를 교환할 수 있다. UNSPSC를 지원하는 제품분류 자동화 소프트웨어인 GoldenBullet은 UNSPSC의 분류체계와 제품에 대한 설명을 입력받아 간단한 온톨로지로 변환하며, UNSPSC 내에서 유사도가 큰 항목으로 분류한다 (UNSPSC website 2003).

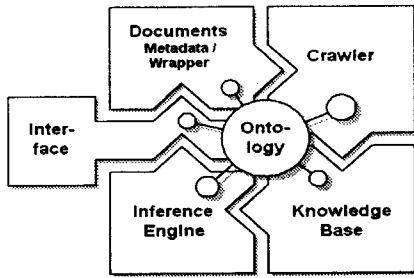
2.3.3 각종 회의정보 제공 시스템: ITTALKS

ITTALKS는 IT 관련 세미나 및 회의에 대한 정보를 종합적으로 제공하는 시스템으로 세미나를 도메인으로 사람과 에이전트간, 에이전트와 에이전트간 상호작용을 가능하게 하는 DAML 기반의 지능형

서비스 제공 시스템이다(Cost et al. 2002).

온톨로지의 상호운영 방법에는 정렬, 제휴의 방법과 통합, 병합 방법이 있다. ITTALKS 시스템은 다양한 온톨로지를 제휴의 방법을 이용해 연결하여 상호운용하고 있는데 제휴의 방법을 사용하면 그룹에서와 같이 온톨로지 간의 독립성이 보장될 수 있다. 본 연구에서도 가구 온톨로지를 도메인으로 하여 몇 개의 복수 온톨로지를 제휴의 방법으로 운용하였다. 제휴방식의 시스템에서 각각의 온톨로지는 끊임없이 수정 발전하여 특정 온톨로지의 개념이 발전하고 확장될 수록 전체적인 시스템도 동시에 성장하게 된다(양정진 2003).

그 밖의 온톨로지 시스템 관련 연구로는 EU의 여러 연구 기관이 컨소시엄을 구성하여 개발한 기업 내 지식정보 관리를



〈그림 6〉 Ontobroker의 개요도

위한 On-To-Knowledge (<http://www.ontoknowledge.org>) 시스템과 Karlsruhe 대학의 AIFB에 의해 추진된 온톨로지 시스템으로, 정보의 통합 및 검색을 위한 프로젝트로 탄생한 Ontobroker (http://ontobroker.aifb.uni-karlsruhe.de/index_ob.html) 등이 있으며, 국내의 연구로는 온톨로지 시스템을 이용해 웹 상의 국제기구 관련 정보의 관리 및 검색시스템 구현을 위한 실험적 연구(김현희, 안태경 2003) 등이 있다.

3 시소러스와 시맨틱 웹

3.1 시소러스와 온톨로지

시맨틱 웹은 풍부한 어휘를 지닌 다양한 온톨로지 시스템 구축을 전제로 구현되는 모형이다. 온톨로지는 일반적으로 클래스와 속성, 그리고 제약사항들(constraints)의 집합으로 구성되는데, 온톨로지의 도메인과 개념범주를 결정하고 클래스(개념)들 사이의 분류체계(taxonomy)를 설정하는 것이 선행되어야 한다. 이러한

면에서 시소러스는 중요한 의미를 가지게 된다. 잘 구축된 기존의 시소러스 개념체계를 온톨로지로 전환하는 것이 매우 효율적인 방법이 될 수 있다는 것이다. 이와 관련해, W3C의 OWL 제안서는 OWL Lite를 통해 웹 응용에 적합하고 시소러스의 온톨로지 변환에 유용한 기능을 제공한다고 밝히고 있으며(Smith, Welty, and McGuinness 2003), 향후 온톨로지 연구에서 중요한 부분을 차지할 것으로 예상되는 것은 기존의 시소러스에 대한 온톨로지 변환연구라 지적한 연구도 있다(오삼균 2002). 또한 시소러스의 온톨로지 변환 연구(From Thesaurus to Ontology)에서는 예술작품에 대한 색인어를 제공하는 RDF(S) 기반의 시스템 구현을 위해 기존의 시소러스 개념체계를 도입하였다(Wielings et al. 2001). 본 연구에서도 온톨로지를 구축하는 과정에 기존의 시소러스의 도입방법을 택하였고 패싯형 시소러스인 Art and Architecture Thesaurus를 선정하였다.

3.2 Art and Architecture Thesaurus(AAT)

AAT는 예술·건축과 관련된 서지·그림 데이터베이스 안에서, 또는 해당분야의 소장품을 취급한 문헌에서 다루어진 표준용어(Standardized Vocabulary or Terms)를 집대성한 시소러스이다. 23개의 계층구조를 갖춘 초판(1990년 간행) 이후, 현재는 33개 계층에 28,000개의 우선어와 동의어·관련어를 포함한 총 120,000

개의 용어를 수록하고 있다. 또한 Art and Architecture Thesaurus Online (<http://www.getty.edu/research/tools/vocabulary/aat/>)도 이용할 수 있다. 이 AAT는 다음과 같은 7개의 기본 패킷으로 구성되어 있다.

- ① Associated Concepts Facets: 개념
- ② Physical Attributes Facets: 물리적 속성
- ③ Styles and Periods Facets: 양식과 시대
- ④ Agents Facets: 소장자나 소유주, 개인 또는 단체
- ⑤ Activities Facets: 행위, 물질·정신적 활동 및 학술분야
- ⑥ Materials Facets: 재료, 물질적 특성
- ⑦ Objects Facets: 대상자료, 사물

3.3 패킷형 시소러스와 온톨로지 시스템

패킷형 시소러스의 개념 체계는 클래스와 속성으로 표현이 가능하며, 온톨로지 시스템과 패킷형 시소러스의 개념체계가 유사성이 많아 시소러스를 이용하여 온톨로지 시스템의 도메인 선정하는 방법이 효율적일 수 있다.

본 연구에서는 체계적으로 잘 정의된 패킷형 시소러스를 선정하는 것이 무엇보다 중요하였다. 이 과정에서 MeSH(Medical Subject Heading)와 같은 전문적인 분야보다는 AAT같은 일반적인 예술, 건축분야(특히, 가구분야)의 패킷개념이 이용자들에게 익숙할 것이며 비교적 용어간의

개념관계가 명확하게 파악하는데 용이할 것이다. 온톨로지 구축시 가장 중요한 것이 그 범위의 설정과 개념을 구축하고 제어하는 방안이라고 볼 때, 되도록이면 시소러스의 개념체계를 그대로 유지하는 것이 바람직하다고 판단하였다. 새로운 개념체계를 구축하는 것이 아니라, 기존의 개념체계를 그대로 온톨로지화하여 시맨틱 웹의 기반이 되는 온톨로지 시스템을 구현하는 것이 본 연구의 목적이다.

3.4 온라인 시소러스와 온톨로지 시스템

온라인 시소러스와 온톨로지 시스템은, 고유의 도메인과 개념체계를 가지고 있으며 온톨로지 시스템 중 경우에 따라서는 개념체계를 이용자에게 제시한다는 면에서 볼때 매우 유사한 면을 가지고 있다. 또한 시소러스와 온톨로지가 고유의 분류체계를 가진다는 면에서 그 역할에 대해 혼란을 일으킬 소지가 있다. 그러나, 온톨로지 시스템은 독립된 개념체계이면서 동시에 온톨로지간의 상호매핑과 연계를 기본으로 하며, 시맨틱 웹의 추론환경을 위한 기계가독형의 표준화된 포맷이라는 점에서 일반적인 온라인 시소러스와 차이점을 갖는다. 앞에서 언급한 바와같이 온톨로지의 구축을 위해 시소러스를 도입한 것일 뿐, 근본적인 존재의 목적이 다르기 때문에 각각의 역할이 결코 동일하거나 유사하지는 않다. 간단히 말하자면, 시맨틱 웹을 구현하기 위한 기반 시스템이라는 측면에서 온톨로지 시스템을 보는 것

이 적절할 것이다. 본 시스템은 간단한 프로그래밍을 통해 문서파싱을 하여 개념 체계를 브라우징할 뿐 완전한 인터페이스를 제공하지 못하고 있으며, 실제적인 추론시스템이 존재하지 않는 현시점에서 axiom으로 대변되는 온톨로지 기반의 시스템이 시소러스를 비롯한 기존의 검색 시스템에 비해 두드러진 특징을 나타내는 힘든 것이 사실이다. 또한 현재 시점에서 온톨로지 언어로 구현된 개념시스템을 거의 찾아볼 수 없기 때문에 그 연계를 통한 시맨틱 웹 구현을 기대하기 어려운 단계이다. 따라서, 현 단계에서는 광범위한 분야에 걸쳐 다양한 온톨로지 시스템이 구축되는 것이 중요하며, 이를 토대로 기계가독을 통해 추론이 가능한 단계에서 많은 온톨로지 시스템의 연계과정을 통해 거대한 의미 네트워크로서의 힘을 발휘할 수 있을 것이다.

4 시스템 구현

4.1 개념관계 구현

4.1.1 기본관계 정의

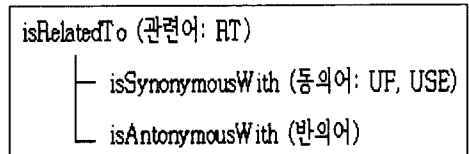
AAT 시소러스의 각 패킷을 온톨로지 언어의 클래스로 기술하였으며, 용어간의 기본개념 관계는 지금까지 일반적으로 시소러스에서 인정되고 있는 용어간의 개념관계인 상하관계, 동의관계, 연관관계를 중심으로 구현하였다.

1) 상위/하위어(BT, NT)

rdfs:subClassOf로 용어 클래스간의 계층관계가 성립하며 트리구조의 부모노드를 상위어, 자식노드를 하위어로 인식한다.

2) 관련어(RT)

동일 계층의 클래스들, 즉 부모를 공유하는 형제어들(siblings)을 관련어로 취급하며, 반의어 및 동의어를 관련어 범주에 포함시킨다. 다만 반의어는 시소러스에 명시되어 있지 않으므로 속성만 제시하였다.



3) 사용자 정의속성 선언(관련어 구현)

```
<owl:ObjectProperty rdf:ID="isRelatedTo">
```

```
  <rdf:type rdf:resource="&owl:SymmetricProperty"/>
```

```
  <rdfs:domain rdf:resource="&rdfs:Class"/>
```

```
  <rdfs:range rdf:resource="&rdfs:Class"/>
```

```
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="isSynonymousWith">
```

```
  <rdfs:subPropertyOf rdf:resource="isRelatedTo"/>
```

```

<rdf:type rdf:resource="&owl:Sym
metricProperty"/>
<rdfs:domain rdf:resource="&rdfs:
Class"/>
<rdfs:range rdf:resource="&rdfs:Cl
ass"/>
</owl:ObjectProperty>

```

... 이하 생략 ...

4) 클래스 선언(armchairs에 대한 동의어 클래스의 예)

```

<owl:Class rdf:ID="arm chairs">
<rdfs:isSynonymousWith rdf:resourc
e=" #armchairs"/>
</owl:Class>

```

```

<owl:Class rdf:ID="armed chairs">
<rdfs:isSynonymousWith rdf:resourc
e=" #armchairs"/>
</owl:Class>

```

... 이하 생략 ...

4.1.2 AAT의 안내어(Guide Term) 처리

다음의 예는 안내어의 계층구조(furniture 이하 계층의 주요 안내어) 중 일부이다. <single seating furniture> <furniture by form or function> 등은 용어개념을 명확히 하며 개념 분류기준을 이용자에게 제시하게 된다. 본 연구에서는 안내어를 모두 클래스로 처리할 경우 패킷 용어와 안

내어의 역할 구별이 모호해지는 문제가 발생하므로 안내어를 사용자정의 속성으로 처리하였다. 다음은 AAT의 안내어 부분을 속성으로 정의하고 하위 클래스로 추이하는 과정을 보여주는 예이다.

```

<furniture by form or function>
  <single seating furniture>
    <chairs by form>
    <chairs by function>
    <chairs by location or context>
    <chairs by design>
    <stools by form>
    <stools by function>
    <stools by location or context>
    <stools by design>
    <folding stools>
    <multiple-seating furniture>
    <sleeping and reclining furniture>
    <storage and display furniture>
    <support furniture>
    <furniture by location or context>
    <furniture by design>

```

```

<owl:Class rdf:ID="seating furniture"/>
<owl:Class rdf:ID="chairs">
  <rdfs:subClassOf rdf:resource=" #seati
ng furniture"/>

```

...
</owl:Class>

```

<owl:ObjectProperty rdf:ID="SingleSeatin
gFurniture">

```

```

  <rdfs:domain rdf:resource=" # .... "/>
  <rdfs:range rdf:resource=" #seating
furniture"/>
</owl:ObjectProperty>

```

```

<owl:ObjectProperty rdf:ID="ChairsByForm">
  <rdfs:subPropertyOf rdf:resource=" #
SingleSeatingFurniture"/>

```

```

    <rdfs:range rdf:resource="#chairs"/>
    ...
</owl:ObjectProperty>
    이로써 다음의 관계가 형성된다.
seating furniture SingleSeatingFurniture
chairs
chairs ChairsByForm armchairs
    
```

개념의 상속과 추이를 통해 다음과 같은 관계도 성립된다.

```

seating furniture SingleSeatingFurniture
armchairs
seating furniture ChairsByForm armchairs
    
```

4.1.3 Axiom을 이용한 의미 확장

온톨로지 언어를 이용해 개념(클래스 및 속성) 간의 관계를 설정하면 명시적으로 표현되지 않은 개념간의 관계에 대한 추론이 가능하다. 이하는 본 연구에서 부가적으로 제시할 수 있는 추론가능 부분을 예로 제시한 것이다.

1) disjoint(OWL DL)

일인용 가구(chair)와 다인용 가구(bench, sofa, wagon seat 등)를 disjoint로 정의하면

```

<owl:Class rdf:ID="chairs">
    <rdfs:subClassOf rdf:resource="#seating furniture"/>
    <owl:disjointWith rdf:resource="#benches"/>
    
```

```

    <owl:disjointWith rdf:resource="#sofas"/>
    <owl:disjointWith rdf:resource="#wagon seats"/>
</owl:Class>
    
```

일인용 가구의 서브클래스와 다인용 가구의 서브클래스는 서로 disjoint 관계로 정의되며, 따라서 둘을 공통으로 하는 하위클래스는 존재할 수 없다. 따라서 의자(일인용)와 벤치(다인용)를 동시에 부모로 갖는 하위클래스를 시스템에서 생성할 경우 시스템은 에러를 추론해 시스템 운용의 오류를 제공할 수 있다.

2) inverseOf

두개의 속성 간에 역속성을 명시하는 이유는 쌍방향으로 정보를 저장하는 것이 중복성을 갖기 때문이며, RDF를 비롯한 온톨로지에서는 중복성을 배제해야 올바른 추론이 가능하다. 일반적인 지식 시스템에서는 역의미관계를 양쪽에 모두 표현해 주어야 하지만, 온톨로지를 이용하면 이 과정을 axiom을 통해 자동화시킬 수 있다(Smith, Welty, and McGuinness 2003). 예를 들어 제품(가구)과 생산자(회사)를 표현할 경우 다음과 같이 표현할 수 있다.

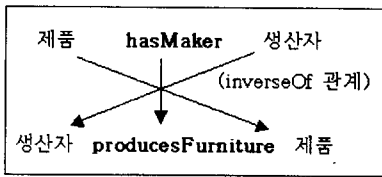
```

<owl:ObjectProperty rdf:ID="hasMaker">
    <rdf:type rdf:resource="#owl:FunctionalProperty"/>
    
```

```

</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="producesFurniture">
  <owl:inverseOf rdf:resource="#"#hasMaker"/>
</owl:ObjectProperty>
  
```

hasMaker를 producesFurniture와 inverseOf 속성으로 정의하면 두 속성간의 관계가 하나의 관련개념으로 설정되어 클래스간의 표현을 hasMaker로만 표현해도 자동으로 hasMaker이면서 역으로 producesFurniture의 관계를 가지게 된다.



4.2 보조 온톨로지를 이용한 시스템 확장

4.2.1 더블린 코어 메타데이터와 VRA (Visual Resources Association) 핵심 범주

VRA 핵심 범주(Core Categories Version 3.0)는 이미지와 시각예술 작품들을 기술하기 위한 레코드의 집합으로서, 작품과 이미지 정보 등 시각자원 정보의 공유를 목적으로 한 것이다. 17개의 정의요소와 한정어들은 표준 메타데이터인 더블린코어와 매핑이 가능하며, AAT의 패킷과도 일부 참조가 가능하다. VRA에서는 데이

터 값의 표준을 제공하는 제어어휘집으로 AACR(Anglo-American Cataloguing Rules), AAT, CDWA(Categories for the Description of Works of Art), LCSH(Library of Congress Subject Headings) 등을 권고하고 있다.

더블린 코어(Dublin Core element set) 메타데이터는 시맨틱 웹의 기반구조의 중요한 부분이다. 자원기술을 위한 공통된 틀을 제공하기 위한 목적으로 제정되어 전 세계적의 다양한 분야로부터 폭넓은 지원을 받고 있다. 본 연구에서의 메타데이터는 온톨로지로 구축된 어휘체계를 담은 컨테이너로서의 역할을 지닌다. 가구에 대한 구체적인 사례(인스턴스 값)를 표현하기 위해 메타데이터의 각 요소를 선정하였고, 이 요소에 대해 온톨로지 시스템이 값을 제공하는 방식을 취하고 있다(<그림 7> 참조).

다음의 <표 2>는 VRA의 요소정의와 한정어를 더블린코어 요소와 매핑시키는 방법을 보여주고 있다. VRA의 일부 요소(Measurements, Material, Technique)는 더블린 코어의 Format 요소와 중복되고 있으며, 또한 더블린 코어의 요소(Creator, Contributor)가 VRA의 Creator 요소에 중복 교차되는 형태를 보이고 있다. 이러한 관계를 표현하기 위해, 다음의 <표 3> '인스턴스 데이터를 위한 RDF 문서 변환용 템플릿'과 같은 기본 문서구조를 만들고, 이를 '인스턴스 데이터 RDF

문서에 대한 RDF 스키마와 같은 클래스

〈표 2〉 인스턴스 입력시스템을 위한
메타데이터 요소와 데이터유형

VRA	Dublin Core	AAT 패시
레코드 유형	타입(Type)	
Type	Type	
Title .Variant .Translation .Series .Larger Entity	Title	
Measurements .Dimensions .Format .Resolution	Format	
Material .Medium .Support	Format	○
Technique	Format	○
Creator .Role .Attribution .Personal name .Corporate name	Creator, Contributor	
Date .Creation .Design .Beginning .Completion .Alteration .Restoration	Date, Coverage	
Location .Current Site .Former Site .Creation Site .Discovery Site .Current Repository .Former Repository	Contributor, Coverage	
ID Number .Current Repository .Former Repository .Current Accession .Former Accession	Identifier	
Style/Period .Style .Period .Group .School .Dynasty .Movement	Coverage, Subject	○
Culture	Coverage	△
Subject	Subject	△
Relation .Identity .Type	Relation	
Description	Description	
Source	Source	
Rights	Rights	

와 속성간의 관계로 정의하였다.

데이터베이스에 입력된 객체에 대한 인스턴스는 다시 변환용 RDF 템플릿에 매칭과정을 통해 새로운 객체 데이터를 생성하게 되고 이를 응용시스템에 제공할 수 있도록 하였다.

4.2.2 보조 온톨로지 시스템 도입

시맨틱 웹이 다양한 온톨로지의 연계로 기반으로 한다는 면에서, 본 연구에서도 소규모의 시맨틱 웹 환경을 위해 다양한 온톨로지를 도입할 필요가 있었으며 이는 시맨틱 웹 환경에서의 복수 온톨로지 운용의 예를 제시하기 위한 것이다.

AAT의 여러 패시 중 몇가지를 도입하여 보조 온톨로지 시스템을 구축한 후 이를 연계하여 운용하는 제휴의 방식을 택하였다. 가구 온톨로지 시스템에서 선택된 개념을 지식화하기 위한 인스턴스 부여를 위해 색상, 재료, 시대와 스타일 온톨로지 분야에서 보조 온톨로지를 구축하였다. 앞서 제시한 VRA 메타데이터 44개 요소에 대해 모두 온톨로지를 제공하지 못한 것은 현 연구단계에서의 한계점이었다. 가구 온톨로지를 비롯하여 각각의 온톨로지 시스템은 독립적이며, 개별적인 발전과정을 통해 전체 시스템의 완성도가 높아질 것이다.

〈표 3〉 인스턴스 데이터 RDF 문서에 대한 RDF 스키마와 인스턴스 데이터를 위한 RDF문서 변환용 템플릿

<pre> <?xml version='1.0' encoding='ISO-8859-1'?> <!DOCTYPE rdf:RDF [<!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'> <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>]> <rdf:RDF xmlns:rdf="&rdf;" xmlns:rdfs="&rdfs;"> <!-- 클래스 선언 --> <rdf:Description rdf:ID="dc:contributor"> <rdf:type rdf:resource="&rdfs:Class"/> <rdfs:subClassOf rdf:resource="&rdfs:Resource"/> </rdf:Description> <rdf:Description rdf:ID="dc:coverage"> <rdf:type rdf:resource="&rdfs:Class"/> <rdfs:subClassOf rdf:resource="&rdfs:Resource"/> </rdf:Description> <!-- 이하생략 --> <!-- 속성선언 --> <rdf:Description rdf:ID="vra:creator.attribution"> <rdf:type rdf:resource="&rdf:Property"/> <rdfs:domain rdf:resource=" #dc:contributor"/> <rdfs:domain rdf:resource=" #dc:creator"/> <rdfs:range rdf:resource="&rdfs:Literal"/> </rdf:Description> <rdf:Description rdf:ID="vra:creator.corporatename"> <rdf:type rdf:resource="&rdf:Property"/> <rdfs:domain rdf:resource=" #dc:contributor"/> <rdfs:domain rdf:resource=" #dc:creator"/> <rdfs:range rdf:resource="&rdfs:Literal"/> </rdf:Description> <!-- 이하생략 --> </rdf:RDF> </pre>	<pre> <?xml version="1.0"?> <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:vra="http://lis.yonsei.ac.kr/~heon/ontology/vra#"> <rdf:Description about="객체의 URI"> <dc:type> <vra:recordtype/> <vra:type/> </dc:type> <dc:title> <vra:title.variant/> <vra:title.translation/> <vra:title.series/> <vra:title.largerentity/> </dc:title> <dc:format> <vra:measurements.dimensions/> <vra:measurements.format/> <vra:measurements.resolution/> <vra:material/> <vra:technique/> </dc:format> <dc:creator> <vra:creator.role/> <vra:creator.attribution/> <vra:creator.personalname/> <vra:creator.corporatename/> </dc:creator> </rdf:Description> <!-- 이하생략 --> </rdf:RDF> </pre>
---	---

4.3 온톨로지 시스템 구현

4.3.1 시스템 개요와 구성요소

1) 시스템 개요

2) 시스템 구성요소

- 주 온톨로지 시스템
 - : 가구 개념 검색 및 계층관계 브라우징, 클래스·속성·관련어 정보 검색
- 보조 온톨로지 시스템
 - : 인스턴스 입력을 위한 제어된 어휘 제공 (색상 온톨로지, 재료 온톨로지, 시대 및 스타일 온톨로지 등)
- 데이터베이스 관리시스템
 - : 질의 개념에 대한 인스턴스 값의 저장
- 온톨로지 변환 시스템
 - : 템플릿의 요소에 해당 인스턴스 값을

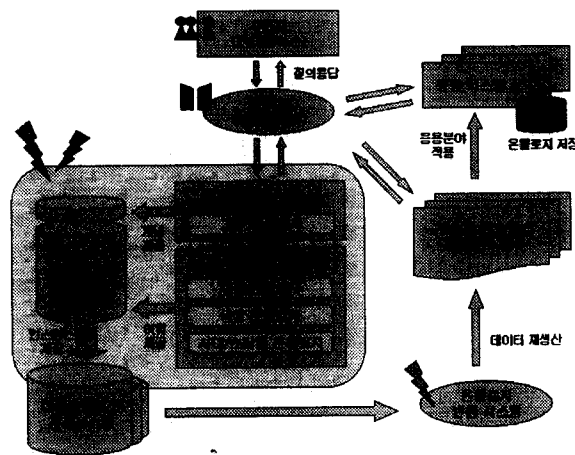
을 치환하여 RDF 데이터를 재생산함

- 웹서버 및 게이트웨이
 - : 사용자와의 인터페이스와 응용 프로그램의 기능 수행
- 응용 시스템 (구현하지 않음)
 - : 온톨로지 시스템을 통해 생성된 RDF 문서를 응용시스템의 데이터로 이용

4.3.2 질의과정

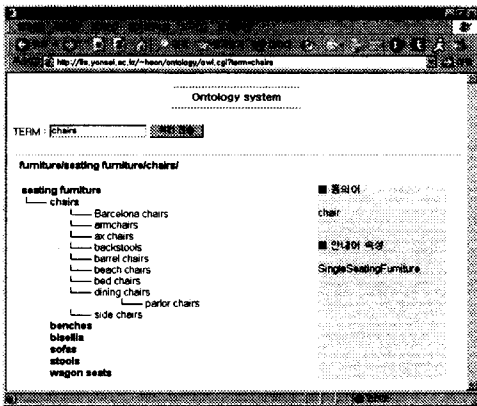
1) 가구개념 검색

: 온톨로지 시스템의 사용자 인터페이스(<http://lis.yonsei.ac.kr/~heon/ontology>)를 통해 가구관련 용어를 질의한다. 개념트리는 상위 1단계, 하위 2단계까지 보여주도록 하였다. 개념트리에서 형제어와 상하위어 등의 관련어를 얻을 수 있고 동의어 리

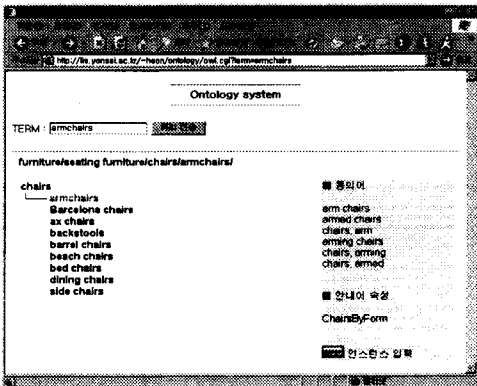


<그림 7> 시스템 구성도

스트와 개념의 분류기준 속성을 얻을 수 있다. 동의어에 대해 우선어를 입력하게 되는데, 예를 들어 armchairs에 대해 arm chairs나 armed chairs 등의 어휘를 입력해도 우선어(Preferred term)인 armchairs로 제어되도록 하였다. OWL문서를 통해 개념트리(<그림 5>), 속성 및 동의어 리스트(<그림 6>) 등은 자동으로 갱신되도록 하였다.



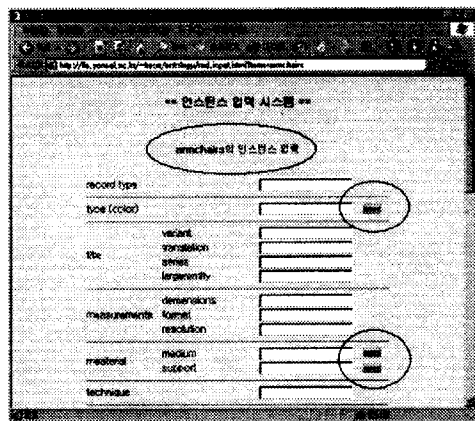
<그림 8> 개념트리



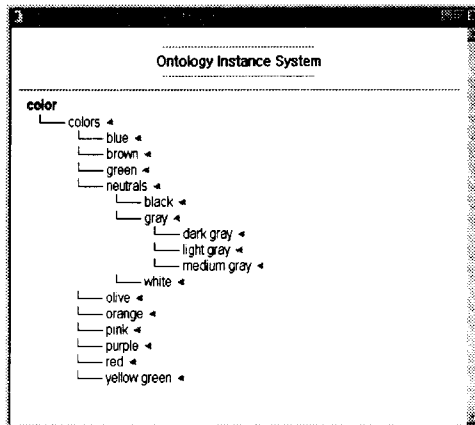
<그림 9> armchairs의 동의어 리스트

2) 개념의 인스턴스 입력

: 이용자가 개념의 인스턴스 입력 버튼을 선택하면 메타데이터 요소 입력창이 나타난다. 보조 온톨로지가 구축되어 있는 요소들은 우측의 INPUT 버튼을 통해 새로운 온톨로지 화면을 제공받는다.



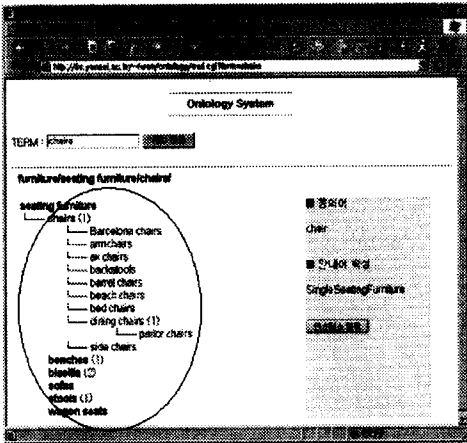
<그림 10> 인스턴스 입력 시스템



<그림 11> Color 온톨로지를 통한 인스턴스 제공

3) 객체에 대한 온톨로지 문서(RDF) 재생산

: 인스턴스가 입력되면 시스템의 개념 트리에 숫자가 나타나 개념의 사례(인스턴스)의 개수를 확인할 수 있다. 입력된 인스턴스 데이터는 응용 애플리케이션의 데이터로 사용하기 위해 온톨로지 변환 시스템을 거쳐 RDF 문서로 재생산될 수 있다.



<그림 12> 개념트리에 제시된 인스턴스 수

The screenshot shows an RDF document in XML format. It starts with a namespace declaration: 'chairs: <http://www.ontology.org/ontology/chairs/>'. The document describes a chair with various properties: 'chairs:chairs' (URI), 'chairs:label' (value: 'chairs'), 'chairs:isA' (value: 'chairs:furniture'), 'chairs:hasPart' (value: 'chairs:backrest'), 'chairs:hasPart' (value: 'chairs:seat'), 'chairs:hasPart' (value: 'chairs:legs'), 'chairs:hasPart' (value: 'chairs:armrests'), 'chairs:hasPart' (value: 'chairs:base'), 'chairs:hasPart' (value: 'chairs:padding'), 'chairs:hasPart' (value: 'chairs:upholstery'), 'chairs:hasPart' (value: 'chairs:finish'), 'chairs:hasPart' (value: 'chairs:hardware'), 'chairs:hasPart' (value: 'chairs:accessories'), 'chairs:hasPart' (value: 'chairs:repairs'), 'chairs:hasPart' (value: 'chairs:cleaning'), 'chairs:hasPart' (value: 'chairs:storage'), 'chairs:hasPart' (value: 'chairs:display'), 'chairs:hasPart' (value: 'chairs:use'), 'chairs:hasPart' (value: 'chairs:maintenance'), 'chairs:hasPart' (value: 'chairs:history'), 'chairs:hasPart' (value: 'chairs:evolution'), 'chairs:hasPart' (value: 'chairs:design'), 'chairs:hasPart' (value: 'chairs:manufacture'), 'chairs:hasPart' (value: 'chairs:distribution'), 'chairs:hasPart' (value: 'chairs:marketing'), 'chairs:hasPart' (value: 'chairs:sales'), 'chairs:hasPart' (value: 'chairs:customer service'), 'chairs:hasPart' (value: 'chairs:warranty'), 'chairs:hasPart' (value: 'chairs:returns'), 'chairs:hasPart' (value: 'chairs:complaints'), 'chairs:hasPart' (value: 'chairs:reviews'), 'chairs:hasPart' (value: 'chairs:ratings'), 'chairs:hasPart' (value: 'chairs:recommendations'), 'chairs:hasPart' (value: 'chairs:reviews'), 'chairs:hasPart' (value: 'chairs:ratings'), 'chairs:hasPart' (value: 'chairs:recommendations').

<그림 13> 변환 시스템의 RDF 문서변환 결과

4.3.3 시스템 구현 환경

온톨로지 언어를 처리하는 시스템 기반으로 리눅스(Redhat Linux 9.0)를 이용하였고, 아파치 웹서버 환경을 사용하였으며, 온톨로지 문서를 처리하기 위해 XML 파서인 expat Python 모듈을 이용하였다. expat API는 SAX 방식으로 특정태그를 만나면 이것을 클래스와 속성 등으로 분리해 낸다. CGI 작성에는 Python과 PHP를 이용하였고 데이터베이스 시스템(DBMS)은 MySQL을 사용하였다.

5 결론

시맨틱 웹은 현재 웹 컨소시엄(W3C)의 OWL을 중심으로 온톨로지 언어에 대한 표준안의 개발을 위한 연구가 활발하게 진행되고 있다. 본 연구에서는 기존의 메타데이터, XML(S), RDF(S)의 개념을 포괄적으로 다루면서 OWL을 중심으로 웹 기반의 온톨로지 시스템을 구축하였다.

Art and Architecture Thesaurus의 Furniture이하 패시을 가구 분야의 온톨로지 시스템으로 구현하는 방법론을 제시하였으며, 응용 시스템으로 확장하는 방안으로 메타데이터 요소와 여러 개의 보조 온톨로지 도입을 추가적으로 고려하였다.

시스템 구축과정에서, 시소러스의 선정과 어휘체계에 대한 분석 과정이 필요하였으며 이는 시스템의 도메인을 결정하는데 가장 중요한 요소였다. 또 온톨로지

언어와 그 제반 기술에 대한 이해와 분석한 어휘체계를 적절하게 구현하는 능력이 요구되었다. 향후 온톨로지 언어를 위한 편집도구가 많이 제작되어 관련 작업에 도움을 줄 수 있을 것으로 보인다. 이밖에 웹 서비스를 위한 서버관리, XML 처리를 위한 파싱 작업과 인터페이스 제작을 위한 프로그래밍 기술이 필요하였고, 전반적인 데이터베이스 시스템의 조장이 필요하였다.

또한, 온톨로지 시스템 구축과정에서 몇가지 문제점과 한계가 드러났는데, 현재 시스템의 도메인이 가구분야에 한정되어 있음에도 불구하고 AAT에서 제시하고 있는 많은 어휘를 충분히 구현하지 못하였다는 점이다. 이는 향후 시스템의 인터페이스를 통해 웹상에서 개념을 입력하여 OWL 문서를 자동 갱신할 수 있도록 발전시키는 것이 바람직 할 것이다. 또한, AAT의 여러 패킷개념을 이용한 보조 온톨로지 역시 더욱 다양하게 구축되어야 하며, 풍부한 어휘를 구축하는 것이 필수적인 조건이라 하겠다. 시스템적인 면에서는 동의어에 대한 우선어 통제는 잘 구현되었으나 안내어 속성의 시스템 내 역할이 명확하지 못하였는데, 안내어의 상속관계 등을 보여주도록 응용처리할 시스템 기능을 구비하지 못한 점은 현 시스템의 한계라 할 수 있겠다. 또한, 본 시스템을 통해 생산된 RDF데이터를 응용할 수 있는 연계분야의 연구 역시 향후 발전시켜야 할 부분이라고 하겠다. axiom을 포

함한 온톨로지의 많은 부분이 향후 개발될 추론시스템에 많은 부분을 의존하고 있으므로, 가시적인 시스템의 성능을 보여주기 힘든 부분도 현 시점에서의 한계라 할 수 있겠다.

결론적으로, 본 연구는 실제 구현 및 응용연구가 부족했던 온톨로지 분야에서 선형연구로써 의미를 가지며, 온톨로지 언어로 시스템을 구축하기 위한 기존의 방법론 즉, 메타데이터를 통합하는 역할로써의 온톨로지(MusicBrainz 시스템), 분류체계 시스템으로써의 온톨로지(UNSPSC 시스템), 복수 온톨로지 시스템의 제휴응용(ITTALKS 시스템) 등과는 또 다른 방식인 온톨로지를 이용한 인스턴스 입력 시스템의 구현사례를 제시했다는 점이 연구의 의의라 할 수 있겠다.

그밖에도 관련 문헌 및 데이터를 검색하고 적합성 여부를 측정하는 온톨로지 기반의 검색시스템 성능평가 연구, 다양한 시소러스의 온톨로지 변환연구는 시맨틱 웹의 기반구축이라는 측면에서 향후 관련 연구에서도 매우 중요한 의미를 지닐 것으로 기대된다.

참 고 문 헌

- 김이란. 2001. 『온톨로지의 의미정보를 이용한 RDF 스키마 생성에 관한 연구』. 석사학위논문, 연세대학교 대학원 문헌정보학과.
- 김태수. 1999. 『목록의 이해』. 서울: 한국

- 도서관협회.
- 오삼균. 2002. 시맨틱웹 기술과 활용방안. 『정보관리학회지』, 19(4): 297-319.
- 최석두. 2002. 『한글 시소러스 구축 기준에 관한 연구』. 박사학위논문, 연세대학교 대학원 문헌정보학과.
- 최중민, 조성정, 김진형, 이재호, 양정진, 김인철, 강민구, 박영택. 2003. <특집: 시맨틱 웹>. 『정보과학회지』, 21(3): 3-50.
- Berners-Lee, T., J. Hendler, and O. Lassila. 2001. "The Semantic Web", Scientific American.
- Brickley, D., and R. V. Guha. 2000. "Resource Description Framework (RDF) Schema Specification 1.0. W3C Candidate Recommendation 27 March 2000". [online]. [cited: 2002. 10. 20]. <<http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>>.
- Broekstra, J., M. Klein, S. Deckter, D. Fensel, F. van Harmelen, and I. Horrocks. 2001. "Enabling Knowledge Representation on the Web by Extending RDF Schema". [online]. [cited: 2003. 4. 10]. <<http://www.cs.vu.nl/~mcaklein/papers/WWW01.pdf>>
- Fensel, D., F. van Harmelen, I. Horrocks, D. L. McGuinness, and P. F. Patel-Schneider. 2001. "An Ontology Infrastructure for the Semantic Web". [online]. [cited: 2003. 4. 5]. <<http://www.cs.vu.nl/~frankh/postscript/IEEE-IS01.pdf>>.
- Fensel, D., I. Horrocks, F. van Harmelen, S. Decker, M. Erdmann, and M. Klein. 2000. "OIL in a Nutshell". [online]. [cited: 2003. 1. 20]. <<http://www.cs.vu.nl/~ontoknow/oil/down/oilnutshell.pdf>>.
- Fensel, D. 2001. "Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce". Springer-Verlag.
- Gruber, T. R. 1993. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing". [online]. [cited: 2002. 12. 20]. <<http://www.cise.ufl.edu/~hammer/classes/6930/XML-FA02/papers/gruber93ontology.pdf>>.
- Hjelm, J. 2001. "Creating the Semantic Web with RDF". Indianapolis : John Wiley.
- Lassila, O., and R. R. Swick. 1999. W3C "Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation 22 February 1999". [online]. [cited: 2002. 10. 20]. <<http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>>.

Smith, M. K., C. Welty, and D. L. McGuinness. 2003. "Web Ontology Language (OWL) Guide Version 1.0". [online]. [cited 2003. 3. 20]. <<http://www.w3.org/TR/owl-guide/>>.

Wielinga, B. J., A. Th. Schreiber, J.

Wielemaker, and J. A. C. Sandberg. 2001. "From Thesaurus to Ontology". [online]. [cited: 2003. 3. 20]. <<http://www.swi.psy.uva.nl/usr/Schreiber/papers/Wielinga01a.pdf>>.