제22회 한국정보처리학회 추계학술발표대회 논문집 제11권 제2호 (2004. 11)

OWL 온톨로지 언어로의 HTML문서 변환 시스템

곽현수, 김수경, 김영근, 안기홍
한밭대학교 컴퓨터공학과
e-mail:khs0520@comeudcation.co.kr, sosha@nate.com,
kimsk@hanbat.ac.kr, khahn@hanbat.ac.kr

A Conversion System of HTML Document into OWL Ontology language

Hyoun-Soo Kwak, Su-Kyoung Kim,
Yeong-Geun Kim, Kee-Hong Ahn
Dept of Computer Engineering, Hanbat National University

요 약

텍스트 중심의 현재의 웹은 주로 시각적 효과만을 고려하여 사용되었으므로, 사용자가 원하는 정보를 효율적으로 추출하기에는 많은 문제점을 지니고 있다. 그래서 점차 메타데이타의 개념을 통하여 웹 문서에 시맨틱 정보를 덧붙이고 이를 이용하여 컴퓨터와 사람이 의사소통을 할 수 있는 시맨틱 웹이 제안되었다. 앞으로 의미 중심의 시맨틱 웹으로 발전해 나가기 위해서는 온톨로지의 구현이 필수적으로 요구되는데, 본 논문은 현재 웹에서 사용되고 있는 HTML언어를 재입력하지 않고, 온톨로지 언어 중 하나인 OWL로 자동 변환하는 시스템을 구현하고자 한다. 온톨로지를 사용함으로써 현재의 웹과 비교하여 좋은 잇점은 문서에 대한 의미와 구조를 파악하여 기계가 의미에 따라 정보를 자동 추론을 할 수 있고, 이기종간의 상호운용성을 보장한다. 또한 현재의 웹에서는 많은 문서들이 서로 동일한 내용으로 작성되는 경우가 많은데, 작성된 온톨로지를 공유하고 재사용하여 그에 따르는 시간과 비용을 줄일 수 있다.

1. 서 론

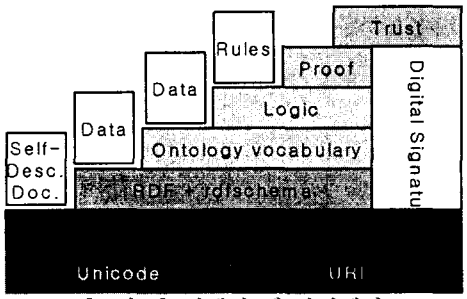
요즘 우리 사회의 모든 생활은 인터넷과 연결됨으로써 네트워크를 통한 신속한 정보 교환 및 지식 공유를 통해 효율적 업무 수행을 할 수 있다. 현재의 웹은 비교적 쉬운 HTML 언어를 사용하기 때문에 누구나 쉽게 정보에 접근할 수 있고, 또 새로운 문서를 웹에 게시할 수도 있다. 그 결과 방대한 양의 정보가 웹에 저장되어 있으나, 이 대부분의 데이타들은 동일한 내용으로 여기저기 흩어져 있을 뿐만 아니라, 단순한 텍스트 형태 또는 비구조적인 형태로 남아있다. 이같이 동일한 내용을 여러 사용자가 만듦으로써, 그에 따르는 비용과 노력, 시간의 낭비를 초래하게 된다. 또한 현재 웹상의 모든 검색 엔진들은 단순한 텍스트만을 가지고 검색을 하기 때문에 관련성이 없는 불필요한 검색 결과가 많이 나타나고 있다. 이같이 HTML은 문서의 내용과 의미를 나타내는 시맨틱 정보를 표현하기 어렵기 때문에, 태그의 유연성을 지원하는 XML을 사용하게 되었다. 이 XML을 기반으로 트리플 구조를 가진 RDF가 메타데이터에 대한 표준 프레임웍으로 개발되었는데, 도메인의 지식을 공유하고 재사용하기 위해서는 이것들로도 충분하지 않았다. 그래서 온톨로지의 필요성이 부각되었으며, 이를 기술할 수 있는 언어 중 OWL은 W3C의 시맨틱 웹 관련 권고안들 중 하나다.

본 연구에서는 HTML 문서를 속성과 클래스의 계층 구조를 가지고 많은 관계를 설정하여 의미를 가질 수 있는 OWL 문서로 자동 변환되는 시스템을 구현하고자 한다. 그리하여 기존의 문서를 다시 만들지 않고, 이를 OWL로 변환하여 시맨틱 웹의 구현에 한걸음 더 다가갈 수 있는 토대를 구축하려고 한다. 본 논문의 구성은 2장에서는 관련연구에 대해 살펴보고, 3장에서는 태그의 구조적 의미에 따른 OWL 구문 정의를 설명하고, 4장에서는 HTML을

OWL로 변환하는 시스템을 구현하는 방법에 대해 설명하였다. 마지막으로 5장에서는 결론 및 향후 연구 방향에 대해 논의하였다.

2. 관련 연구

2.1. 시맨틱 웹



[그림 1]. 시맨틱 웹 아키텍처

시맨틱 웹은 현재의 웹과 별개의 것이 아니라, 현재의 웹에 의미와 구조를 담아 보완하는 것으로 팀 버너스리에 의해 [그림 1]과 같은 계층적 구조로 표현되었다.

웹 자원을 가리키는 어드레싱 방법으로 URI가 밑받침되고, 그 기반위에 사용자가 정의하여 태그 이름을 유연하게 사용할 수 있는 XML이 구축되었다. 이때 태그(element,attribute)의 어휘 혼란을 방지하여 문서를 처리하기 위해 네임스페이스를 기반으로 표현하였으며, 이를 구조적으로 나타내며, 데이터 타입 등을 표현할 수 있도록 XMLschema를 정의하였다. 그 위에 RDF, RDFschema로 자원들에 대한 관계를 정의하고, 속성과 클래스의 계층구조에 대한 의미를 제공한다. 그 상위 레이어인 온톨로지에서는 속성과 클래스에 대하여 기술할 수 있는 더 많은 어휘를 제공한다. 온톨로지는 단어와 관계들로 구성된 일종의 사전으로, 시맨틱 웹의 목적인 의미에 따른 자동 실행과 추론을 하기 위한 기술로 사용되고 있다. 온톨로지는 정보검색시 에이전트가 이용자의 요청에 관련된 정보를 추론하는 지식을 제공하는 역할을 하며, 개념-값, 클래스-인스턴스, 클래스-슈퍼클래스, 부분-전체와 같은 개념적 관계를 제공하는 기반이 된다. 그 기반위에 인공지능의 추론에 대한 Logic에 대한 연구가 진행 중이며, 신뢰성과 보안에 관련되어 proof와 Trust에 대해 앞으로 연구 대상이 되고 있다.[1][2]

2.2 OWL (Web Ontology Language)

웹온톨로지 언어가 개발되기 시작한 이유는 다음과 같은 관계를 지원하는 표현력이 결여된 RDF와 RDF 스키마의 모델링 요소를 확장, 강화할 필요가 있었기 때문이다.

- (비)동치성 - sameAs, differentFrom 등
- 속성의 특성 - inverseOf, Transitive, Symmetric 등
- 속성의 제약 - allValueFrom, someValueFrom, Cardinality 등
- 클래스의 공리 - oneOf, dataRange, disjointWith 등
- 부울조합 - unionOf, complementOf, intersectionOf 등[3]

그 결과 나타난 것이 DAML+OIL인데, OWL은 이 DAML+OWL에 기반을 둔 온톨로지 구축 경험을 토대로 class와 property의 상속적 계층구조의 개념과 그 개념들 사이의 관계가 보다 명료하게 정의되도록 정리한 온톨로지 언어이다.

OWL은 다음과 같이 서로 다른 표현력을 가진 3가지 하위 언어로 구성된다.

- OWL Lite - 클래스 분류 계층과 간단한 제약사항 표현을 필요로 하는 사용자들을 위한 언어
- OWL DL - Computational Completeness(모든 결론이 계산될 수 있다는 특성)와 Decidability (모든 계산이 유한한 시간안에 끝난다는 특성)을 유지하면서 최대의 표현력을 활용하고자 하는 사용자에게 적합
- OWL Full - 최대의 표현력과 RDF의 유연한 문법을 모두 활용하고자 하는 사용자에게 적합

위의 3가지 OWL은 서로 다른 개발자 및 사용자를 대상으로 하기 때문에 어떤 OWL 하위 언어가 주어진 요구사항에 최적인지 결정해야 하며, 이들 언어는 다음과 같은 포함관계를 가진다.

OWL Lite < OWL DL < OWL Full [4]

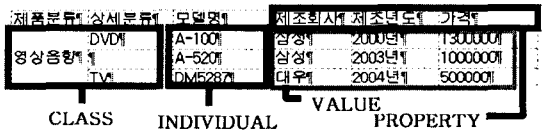
OWL 온톨로지 구성요소를 살펴보면 다음과 같다

- 클래스 간의 택사노미 관계
- 데이터의 속성, 즉 클래스의 요소인 속성값에 관한 기술
- 객체의 속성, 즉 클래스 요소간의 관계
- 클래스들의 인스턴스
- 속성들의 인스턴스 [5]

3. 태그의 구조적 의미에 따른 OWL 구문 정의

우리가 가지고 있는 모든 자료들은 일종의 객체-관계형 데이터베이스의 집합이라고 볼 수 있다. 웹에 있는 대부분의 데이터베이스들은 테이블 구조로 되어있다. 그래서 TABLE 태그를 중심으로 구문을 파악하여 사용자에게 몇가지 사항만을 입력받아 자동으로 OWL로 변환하도록 하는데, 이때 사용되는 태그들을 분석하여 어떤 관계와 구조를 가지고 있으며, 이에 어떤 변환 규칙을 적용하여야 하는지를 살펴보면 다음과 같다.

(1) TABLE 태그는 일종의 데이터베이스 구조로, 필드와 레코드로 구성된다. 이때 필드는 대부분 Property에 해당하며, 레코드는 Individual에 해당된다. Individual의 경우 상하위 개념이 있는 포함구조로 되어 있는 경우가 있는데, 이런 경우는 클래스로 정의된다. 아래 [그림 2]와 같은 웹 문서가 있을 때, 왼쪽 열이 클래스(속성)가 되고, 첫 번째 행이 속성(클래스)이 된다.



[그림 2]. 간단한 웹문서의 테이블 구조

다음은 OWL 규칙에 맞게 class와 property, individual의 관계를 각각 정의한 결과이다. [6][7]

```
영상음향의 하위 클래스인 DVD class 정의 :
<owl:Class rdf:ID="DVD">
 <rdfs:subClassOf>
  <owl:Class rdf:about="영상음향">
 </rdfs::subClassOf>
</owl:Class>
제조회사 property 정의 :
<owl:DatatypeProperty rdf:ID="제조회사">
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:DatatypeProperty>
A-100에 대한 individual 정의 :
<DVD rdf:ID="A-100">
<제조회사
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">삼성</제조회사>
<제조연도
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">2000년</제조연도>
<가격 rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1300000</가격>
</DVD>
```

[그림 3]. Class,Property,Individual을 정의하는 방법

(2) CHECKBOX는 INPUT 태그 안에 기술된 것 중에 개수와는 상관없이 다중 선택을 할 수 있다.

예를들어, "색상선택"이라는 속성이 있을 때, "빨강, 노랑, 파랑"의 세가지 색상 중에서 선택을 해야 하는 CHECKBOX가 있다면 "색상"이라는 클래스의 ID로 "빨강, 노랑, 파랑"을 정의하고, "색상선택"이라는 속성에서는 "색상" 클래스를 참조하도록 하면 된다. 이에 대한 OWL은 다음과 같이 정의된다. [6][7]

```
<owl:Class rdf:ID="색상"/>
<owl:ObjectProperty rdf:ID="색상선택">
   <rdfs:range rdf:resource="#색상"/>
   <rdfs:domain rdf:resource="#제품"/>
</owl:ObjectProperty>
<구매방법종류 rdf:ID="빨강"/>
<구매방법종류 rdf:ID="노랑"/>
<구매방법종류 rdf:ID="파랑"/>
```

[그림 4]. 특정 Class에서 속성값을 정의하는 방법

(3) RADIO 단추는 거의 CHECKBOX와 비슷하지만 CHECKBOX는 다중 선택이 가능한데 반하여 RADIO 단추는 하나만 선택해야 하는 제약사항을 가지고 있다. 이것은 OWL에서 FunctionalProperty라는 속성의 특성을 사용하여 해결할 수 있다. 위의 ObjectProperty의 정의 안에 아래의 내용만 추가하면 된다. [6][7]

```
<rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
```

[그림 5]. 속성이 유일한 값을 갖도록 정의하는 방법

(4) 콤보상자를 표현하는 SELECT 태그는 OPTION 태그에 기술된 값들을 선택할 수 있고, 개수는 multiple 속성의 유무에 따라 하나만 선택해야하는 경우와 다중 선택을 할 수 있는 경우로 나뉜다.

HTML에서 OWL로 변환하기 위해 표현할 수 있는 태그는 이정도로 요약된다. 데이터 타입이나 기타 여러 가지 관계 설정에 대해서는 사용자의 요구를 입력받아 변경하도록 한다.

4. 구현

본 논문에서 제안한 OWL 언어의 자동 변환 시스템은 다음과 같은 순서에 의해 자바로 구현하였다.

(1) HTML언어를 불러오거나, 태그를 화면에서 입력받는다.

(2) 테이블에서 행과 열 중 어떤 부분을 클래스와 속성으로 정의 할 것인지를 선택하게 된다.

(3) 옵션으로 클래스 혹은 속성들에 대한 데이터 타입이나 관계를 정의할 수 있다.

(4) 사용자에 의해 입력받은 값은 JSP에 의해 자바

로 전달된다.

(5) XML 파서인 XERCES를 사용하여 태그사이의 값들을 가져온다.

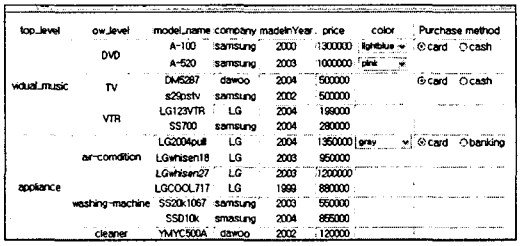
(6) jena를 이용하여 사용자의 입력에 따라, OWL Lite 규칙에 맞게 클래스와 속성을 정의하는 스키마를 구현하고, 그 규칙에 따라 Individual이나 value를 작성한다. [8]

(7) 다음의 두 사이트에서 유효성 검사를 수행한다.

① http://owl.bbn.com/validator/

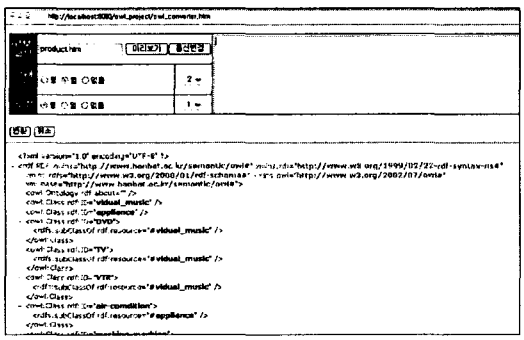
② http://phoebus.cs.man.ac.uk:9999/OWL/Validator

[그림 7]은 [그림 6]과 같은 HTML문서를 OWL로 변환한 결과 화면이다.



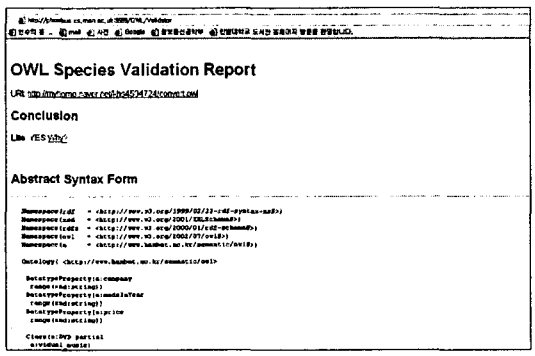
top_level	ow_level	model_name	company	madeInYear	price	color	Purchase method
	DVD	A-100	samsung	2000	1300000	lightblue	⊙card ○cash
		A-520	samsung	2003	1000000	pink	
vidual_music	TV	DM5287	dawoo	2004	500000		⊙card ○cash
		s29pstv	samsung	2002	500000		
	VTR	LG123VTR	LG	2004	199000		
		SS700	samsung	2004	280000		
		LG2004pull	LG	2004	1350000	gray	⊙card ○banking
	air-condition	LGwhisen18	LG	2003	950000		
		LGwhisen27	LG	2003	1200000		
appliance		LGCOOL717	LG	1999	880000		
	washing-machine	SS20k1067	samsung	2003	550000		
		SSD10k	smasung	2004	865000		
	cleaner	YMYC500A	dawoo	2002	120000		

[그림 6] 테이블 구조의 HTML 문서



[그림 7]. OWL로 변환된 결과

[그림 8]는 위에서 생성된 OWL을 Lite영역에서 유효성 검사를 수행한 결과이다.



[그림 8]. 유효성 검사 결과

5. 결론

본 논문에서는 현재의 웹 페이지에서 사용되는 마크업 언어인 HTML을 차세대 시맨틱 웹의 온톨로지에서 사용 될 OWL로 자동 변환되는 시스템에 대해 소개하였다. 이렇게 구현된 온톨로지는 이와 연관된 다른 온톨로지와 병합함으로써 관련된 다른 온톨로지에서 구축된 내용을 재사용할 수가 있다. 예를들어 "영상음향"의 하위클래스로 "TV"라는 클래스가 있는데, 이 "TV"라는 단어에 대한 시소러스 온톨로지와 병합함으로써, "TV"라는 클래스에 대한 구조적 정의 뿐만아니라, 어의 정의까지도 할 수 있다. 그런데 이렇게 다른 온톨로지를 병합할 경우의 문제점은 불일치가 생길 수 있다는 것이다. 어떤 경우에는 "TV"라고 정의를 하지만, 또다른 경우에는 "텔레비젼"이라고 표현을 하는 경우가 있을 수 있다. 또 "TV"가 "영상음향"의 하위 클래스가 아니라 "가전제품"의 하위 클래스가 될 수도 있다. 그러므로 어떤 표준화된 규칙이나, 이를 자동으로 탐지하고 해결할 수 있는 에이전트의 필요성이 요구된다. 이런 온톨로지에 대한 연구가 활발히 진행됨에 따라 추론에 의한 자연어처리 및 기계학습에 대한 방법도 함께 이루어 질 것이다.

자동 실행이나 추론을 할 수 있는 시맨틱 웹으로 발전하기 위해서는 온톨로지는 필수요건이이라 할 수 있으며, 여러 분야의 온톨로지가 구축되고 서로 병합된다면, 앞으로 시맨틱 웹의 행보에 많은 발전을 가져올 것이다.

참고문헌

[1] 최중민. "시맨틱 웹의 개요와 연구동향" 2003.3 정보화학회지 제21권 제3호

[2] Shelley Powers. "practical RDF" O'REILLY.

[3] OWL Web Ontology Language Overview http://www.w3.org/TR/2004/REC-owl-features-20040210/

[4] OWL Web Ontology Language Guide http://www.w3.org/TR/2004/REC-owl-guide-20040210/

[5] 오삼균. "Web Ontology Language와 그 활용에 관한 고찰" 데이터베이스 연구 18권 3호. 2002.9

[6] OWL Web Ontology Language Test Cases http://www.w3.org/TR/2004/REC-owl-test-20040210/

[7] Protege OWL Plugin 2.1 beta

[8] http://www.hpl.hp.com/semweb/jena.htm