

# 온톨로지 객체 모델 생성 시스템 설계

박천수\*, 이미경\*, 손주찬\*, 함호상\*  
\*전자통신연구원 비즈니스 지식처리 연구팀  
e-mail : [bettle@etri.re.kr](mailto:bettle@etri.re.kr)

## Design of Ontology Object Model Generation System

Cheon-Shu Park\*, Mi-Kyoung Lee\*, Joo-Chan Sohn\*, Ho-Sang Ham\*  
\*Business Knowledge Research Team, ETRI

### 요 약

본 논문은 웹 온톨로지 데이터를 접근, 표현 및 처리 할 수 있는 온톨로지 객체 모델을 생성하기 위한 시스템이다. 시맨틱 웹의 대두로 인해 웹 상에 존재하는 데이터의 특성에 따라서 접근 할 수 있는 방법도 다양화 되었다. 이에 웹 상에서 산재되어 있는 지식들을 가져와 각 도메인에 맞게 새로운 온톨로지를 생성하고 서로 다른 언어로 표현된 온톨로지를 계층 어휘들을 이용하여 시맨틱 웹 환경에서 지식을 처리하기 위해 웹 온톨로지를 구축하고 처리할 수 있는 온톨로지 객체 모델을 제공하고, 온톨로지 객체 모델 API 를 통해 외부 어플리케이션과의 정보를 교환한다. 본 논문에서는 웹 온톨로지를 표현하기 위한 모델을 계층별로 구별하여 프레임 기반의 상위 온톨로지(frame-based ontology layer), 다른 도메인에서도 사용이 가능한 공통된 어휘(vocabulary)를 표현한 핵심 온톨로지(generic ontology layer)와 각각의 온톨로지 언어에 의존적인 어휘를 표현한 기능 온톨로지(functional ontology layer)로 구성하여 표현의 중복을 없애고 재 사용성을 높이기 위한 모델을 제공함으로써, 온톨로지 추론, 병합 및 저작 도구 등의 외부 어플리케이션이 온톨로지 객체 모델에 손쉽게 접근할 수 있고, 온톨로지에 대한 쉬운 지식 표현 및 핸들링을 제공할 수 있다.

### 1. 서론

본 시스템은 시맨틱 웹 환경에서 비즈니스 지식을 처리하기 위해 웹 온톨로지를 구축하고 처리 할 수 있는 모델을 제공한다. 웹 상에서 산재되어 있는 지식들을 가져와 각 도메인에 맞게 새로운 온톨로지를 생성하고 서로 다른 언어로 표현된 온톨로지를 처리 할 수 있는 방법을 제공하기 위한 모델을 생성하고 모델에서 제공되는 API 를 통해 온톨로지에 대한 저작, 병합, 추론을 위한 작업을 수행 할 수 있도록 하는 시스템이다.

기존의 웹 환경에서 의미를 가지는 지능화된 웹 환경으로 발전하면서 시맨틱 웹이라는 새로운 형태의 의미를 가지는 웹이 나타나게 되었고, 이를 표현하기 위

한 수단으로 RDF(Resource Description Language)[1], [2]가 표준으로 자리를 잡았고, RDF 로 표현할 수 없는 어휘들을 수용하여 온톨로지들의 병합 및 추론을 위한 방법을 제공하기 위해 현재 OWL(Web Ontology Language)[3]-[7]라는 표준이 진행 중에 있다. 이에 따라서 온톨로지를 표현할 수 있는 언어들에 대한 접근 방법, 처리 방법에 대한 연구가 필요하게 되었다. 현재 진행되고 있는 온톨로지 언어로 표현된 문서를 접근하고 처리하는 방법은 XML 데이터 모델을 이용하여 RDF, DAML-OIL(DARPA Agent Markup Language-Ontology Inference Layer)[8], OWL 에 대한 시맨틱 정보를 표현하고 있다.

본 논문의 구성은 다음과 같다. 2 장에서는 온톨로지 객체모델의 필요성에 대한 내용을 기술하고, 3 장에

서는 온톨로지 객체 모델 생성 시스템의 전체적인 구조, 온톨로지 객체 모델을 생성하기 위한 온톨로지 어휘에 대한 계층적 구조에 대하여 설명한다. 4 장에서는 결론 및 향후 연구 방향에 대해 설명한다.

## 2. 온톨로지 객체 모델의 필요성

온톨로지 기반의 응용들은 의미기반의 정보 접근(semantic-driven information access)이 가능하도록 하는 것이 중요한 이슈가 되어 왔다. 이러한 접근방식의 많은 응용들은 자동화된 정보 처리(automated information processing), 정보 통합(information integration), 또는 지식 관리 등으로 불려왔다. 특히, 팀 버너스리에 의해 기존의 웹이 온톨로지 기반의 메타 데이터에 의해 표현될 수 있는 시멘틱 웹이 새롭게 만들어 지면서 온톨로지에 대한 연구가 증가 되었고, 대량의 정보를 사담에서 자동화된 에이전트를 통해 처리 될 수 있는 방법을 고려하게 되었다. 이러한 온톨로지 모델링의 특성을 해결하기 위해 수많은 온톨로지 언어가 개발되었다. 시멘틱 웹에 대한 기본적인 모델링을 제공하기 위한 언어인 RDF 가 개발되었고, RDF 와 DAML-OIL 을 기반으로 하여 W3C 에서 개발되어 현재 표준화를 진행 중인 OWL 은 RDF 와 DAML-OIL 의 표현력을 확장하고 추론에 있어 효율성을 높이기 위해 만들어 졌으며, FOL(first-order logic)의 부분 집합(subset)으로 만들어진 언어이다. 그러나, 현재 온톨로지 관리(ontology management)와 추론(reasoning)을 향상시키기 위한 상당한 연구에도 불구하고 DB 만큼 활성화 되지 못했다.

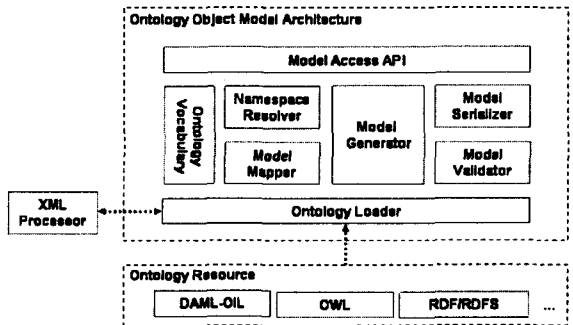
로직 기반(Logic-based)의 접근 방식은 모델에 대한 표현력을 중심으로 초점이 맞추어져 있으므로 인해 다른 시스템과의 통합의 용이성을 제공하지 못하고 처리 성능에 있어서도 한계를 들어냈다. 이러한 문제로 인해 현재 온톨로지 기술 기반의 성공적인 응용 사례가 나오지 못했다. 잘 정의된(well-defined) 시멘틱을 기반으로 메타 개념(meta-concept)을 모델링하고, 모듈화 하는 것이 중요하고, 개념 모델링을 위해 직관적으로 보기에 명료하고 쉽게 시각화 할 수 있으며 기계가 쉽게 이해 할 수 있는 형태의 모델링 방법을 택해야 한다. 또한, 시멘틱 웹의 대두로 인해 웹 상에서 온톨로지 데이터의 특성에 따라서 접근 할 수 있는 방법도 다양화 됨에 따라 현재 진행되고 있는 웹 온톨로지 뿐만 아니라 기존에 구축된 지식베이스 시스템에 접근하여 데이터를 핸들링 할 수 있는 모델이 필요하다. 이에 본 시스템에서는 다양한 온톨로지 언어들을 표현하고 접근할 수 있는 모델을 제시하고, 모델을 기반으로 온톨로지 저작, 온톨로지 추론, 온톨로지 병합과 같은 응용들에서 수행 할 수 있도록 모델 접근 API 를 제공한다.

## 3. 시스템 구조

[그림 1]은 온톨로지를 생성, 접근, 처리 할 수 있는 온톨로지 객체 모델의 구조를 도시 한다. 본 시스템은

웹 온톨로지 언어를 접근하여 이를 분석하여 메모리 상에 모델을 생성하고 외부 어플리케이션에서는 모델 접근 API 를 통하여 모델에 대한 정보를 교환하고, 메모리 상에 생성된 모델에 대한 검증 및 serialize 등을 할 수 있는 기능을 제공하는 모듈로 구성되어 있다.

모델 접근 API 는 온톨로지의 추론, 온톨로지의 병합, 온톨로지 저작 도구 등의 어플리케이션에서 웹 온톨로지 언어에 대한 모델을 접근하고 어플리케이션에서 필요 하는 정보를 제공 할 수 있는 인터페이스 기능을 제공한다.



[그림 1] 온톨로지 객체 모델 생성 시스템 구조

온톨로지를 구성하기 위해 사용할 수 있는 온톨로지 언어는 W3C 에서 표준화를 진행 중에 있으며, 현재 표준화로 진행중인 OWL(Web Ontology Language)와 리소스를 기술하기 위한 표준인 RDF(Resource Description Language), DAML-OIL(DARPA Agent Markup Language-Ontology Inference Layer)등의 온톨로지 언어를 이용해 온톨로지에 대한 문법과 의미를 표현 하고 있다.

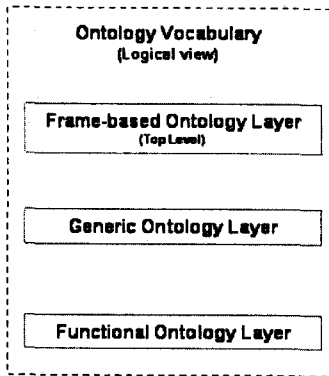
본 논문에서는 이러한 온톨로지 자원을 입력으로 받아서 Ontology Loader 에서 온톨로지 문서에 대한 분석을 하게 된다. Ontology Loader 는 온톨로지 언어를 입력 받아 XML 데이터를 처리할 수 있는 XML 프로세서서를 사용하여 온톨로지 언어를 파싱(parsing)하고, 온톨로지 언어의 특성에 따라 정보를 수집하게 된다. 수집된 정보는 Model Generator 를 통해 전달되게 되며 이때, 각 온톨로지 언어에 따른 Namespace 를 처리 하기 위한 Namespace Resolver 를 거쳐 메모리상에 모델을 생성 하게 된다.

Namespace Resolver 는 모델 생성부에서 온톨로지 객체 모델의 생성시 필요한 네임스페이스를 처리하고, 상기 Model Serializer 에서 온톨로지 객체 모델을 사용자가 인식할 수 있는 파일 형태로 저장 시 네임스페이스를 이용하여 온톨로지 언어를 구별하도록 한다.

Model Generator 는 모델을 생성하기 위해 Ontology Vocabulary 에서 이미 정의된 어휘를 사용하여 입력 받은 온톨로지 언어에 맞는 메모리 상의 온톨로지 객체 모델을 생성하게 된다.

Model Mapper 은 RDF 기반의 온톨로지 언어 외에 또 다른 형태의 지식 표현 언어를 위한 매핑을 위한 모듈로 Ontology Vocabulary 에서 정의 되지 않은 기존의 어휘에서 확장하거나 구현하여 어휘를 정의할 수 있도록 한다. 생성된 모델은 Model Validator 을 통해 모델이 유효한 모델인지를 검증할 수 있는 모듈이다. 이렇게 검증이 끝나면 Model Serializer 을 통해 모델링한 메모리 상의 모델을 원하는 온톨로지 언어로 생성할 수 있다. 온톨로지를 새로 구축하거나 기존의 온톨로지를 불러들여 변경하고자 할 경우에 Model Access API 을 통해 외부 어플리케이션과의 인터페이스를 제공한다.

[그림 2]는 온톨로지를 표현하기 위한 어휘를 정의한 논리적 계층도를 나타낸다.

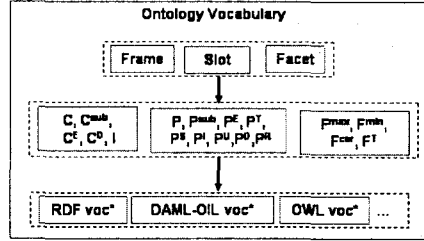


[그림 2] 온톨로지 객체 모델의 논리적 계층도

온톨로지 객체 모델은 프레임 기반(frame-base)의 최상위 온톨로지 계층, 공통된 공리(axioms)와 어휘를 표현하는 공통 온톨로지 계층, 온톨로지 언어에 의존적인 기능 온톨로지 계층으로 구성된다. 이러한 계층 구조를 제공함으로써 각각의 온톨로지 언어에서 표현하고자 하는 어휘의 중복성을 줄이고, 새로운 온톨로지 언어나 기존의 지식베이스 시스템에서 표현된 정보를 상위 온톨로지 계층과 공통 온톨로지 계층을 이용하여 새로운 어휘를 정의 함으로써 확장성을 제공할 수 있을 뿐만 아니라 재사용성을 높일 수 있다.

[그림 3]은 [그림 1]에서 Ontology Vocabulary 의 온톨로지 언어에서 제공될 수 있는 어휘들에 대한 물리적 계층도를 나타낸다.

본 논문에서 제안한 모델에서 지원하는 온톨로지 어휘들을 표현하기 위한 계층에 대한 세부 내용은 다음과 같다. 첫째, 프레임 기반의 Frame-based Ontology Layer 는 Frame, Slot, Facet 의 3 가지로 구성된다. 프레임(Frame)은 하나의 개념에 해당하는 것으로 클래스나 인스턴스(Instance)를 말하며, 슬롯(Slot)은 클래스나 인스턴스에 대한 속성을 나타낸다.



voc<sup>\*</sup> : vocabulary  
 C : class, C<sup>sub</sup> : subClassOf, C<sup>E</sup> : equivalentClass, C<sup>D</sup> : disjointWith, I : instance  
 P : property, P<sup>sub</sup> : subPropertyOf, P<sup>E</sup> : equivalentProperty, P<sup>T</sup> : transitive,  
 P<sup>S</sup> : symmetric, P<sup>I</sup> : inverseOf, P<sup>U</sup> : unique, P<sup>D</sup> : domain, P<sup>R</sup> : range  
 F<sup>max</sup> : cardinality, F<sup>min</sup> : minCardinality, F<sup>max</sup> : maxCardinality, F<sup>T</sup> : type

[그림 3] 온톨로지 객체 모델의 물리적 계층도

패시(Facet)은 슬롯(Slot)에 대한 제약 조건을 나타낸다. 예를 들어 'Author'라는 클래스를 정의하면 Author 는 Name, Age, Address 와 같은 속성들을 가지게 되는 데 이것이 Slot 에 해당된다. Slot 에 대한 제약조건 즉, Slot 에 대한 카디널리티(cardinality) 또는 Slot 값에 대한 데이터 형태(type)를 나타내는 Facet 은 Name 의 형태(Type)이 string 이라는 것을 명시할 수 있다. 표현은 다음과 같다.

Class level

- C:class, C<sup>sub</sup> :subClassOf, C<sup>E</sup> :equivalentClass, C<sup>D</sup>:disjointWith, I : instance

둘째, 온톨로지 계층에서 공통적으로 사용될 수 있고 재활용이 가능한 부분으로 Generic Ontology layer 은 다음과 같은 어휘들로 표현 된다. 우선 클래스 레벨에서 제공되는 특성들로 'Class', 클래스의 상 하위 관계를 나타내는 공리'subClassOf', 동일한 클래스를 정의하는 공리 equivalentClass, 서로 다른 클래스임을 명시하는 공리 disjointWith, 클래스들이 실제 구체화된 객체인 instance 로 구성되며, 속성들에 대한 정보를 나타내는 속성 레벨에서는 속성을 나타내는 property, 속성들간의 상 하위 관계를 나타내는 공리 subPropertyOf, 동일한 속성임을 표현하는 공리 equivalentProperty, 속성의 transitive 관계를 나타내는 공리 transitive, 속성들의 역 관계를 나타내는 공리 inverseOf, 속성들의 symmetric 관계를 나타내는 공리 symmetric, 속성의 고유함을 나타내는 공리 unique, 속성이 속하는 도메인과 범위를 나타내는 domain, range 로 구성된다. 표현은 다음과 같다.

Property Level

- P : property, P<sup>sub</sup> :subPropertyOf, P<sup>E</sup> :equivalentProperty, P<sup>T</sup> :transitive, P<sup>S</sup> :symmetric, P<sup>I</sup> :InverseOf, P<sup>U</sup> :unique, P<sup>D</sup> :domain, P<sup>R</sup> :range

또한 속성들에 대한 제약조건이나 타입을 표현하기 위한 패시 레벨로 다음과 같은 어휘를 표현한다. 속성의 발생횟수를 정의하기 위한 cardinality, 최소 발생횟수를 정의하기 위한 minCardinality, 최대 발생횟수를

정의하는 maxCardinality, 속성의 형태(string, date, decimal 등)를 나타내는 type 으로 구성된다. 표현은 다음과 같다.

Facet Level

-  $F^{car}$  : cardinality,  $F^{min}$  : minCardinality,  
 $F^{max}$  : maxCardinality,  $F^T$  : type

셋째, 온톨로지 언어에서 제공되는 서로 다른 특성들을 표현하는 부분으로 Functional Ontology Layer 은 Frame-based Ontology Layer, Generic Ontology Layer 을 이용하여 표현 되는 계층으로 온톨로지 언어를 구성하기 위한 구성자(constructor)들에 대한 어휘를 나타낸다. Functional Ontology Layer 는 각 온톨로지 언어 (RDF/RDFS, DAML-OIL, OWL)언어에서 제공되는 서로 다른 어휘들을 표현 한다. 예를 들면, OWL(Web Ontology Language)에는 Class 를 구성하기 위한 요소로 Class constructors 가 있다. 클래스 들의 교집합을 나타내는 intersectionOf, 합집합을 나타내는 unionOf, 데이터의 열거를 나타내는 enumeration 형태의 oneOf, 클래스의 차 집합 관계를 나타내는 complementOf 등의 요소가 있다.

4. 결론

본 논문은 웹 온톨로지 데이터를 접근, 표현 및 처리 할 수 있는 온톨로지 객체 모델의 생성 시스템으로, 온톨로지 생성 및 저작, 추론, 병합을 위해 사용할 수 있는 온톨로지 핸들링을 위한 모델을 제공한다. 본 시스템에서 제안한 온톨로지 객체 모델은 기존의 웹 온톨로지에 대한 접근 방법으로 사용되었던 각각의 언어에 맞는 파서(Parser)를 이용하여 메모리 상에 모델을 생성하여 접근하는 방법과는 다르게 웹 온톨로지를 표현하기 위한 모델을 계층별로 구별하여 프레임 기반의 상위 온톨로지(frame-based ontology layer), 다른 도메인에서도 사용이 가능한 공통된 어휘(vocabulary)를 표현한 핵심 온톨로지(generic ontology layer)와 각각의 온톨로지 언어에 의존적인 어휘를 표현한 기능 온톨로지(functional ontology layer)로 구성하여 표현의 중복을 없애고 재 사용성을 높이기 위한 모델을 제공함으로써 온톨로지에 대한 쉬운 지식 표현과 접근 및 핸들링을 제공하도록 하였다.

향후, 본 시스템에서 제안한 모델을 기반으로 현재의 웹 온톨로지 언어에 대한 표현과 접근 및 온톨로지 저작, 추론, 병합에 적용하여 기존의 모델과의 접근의 용이성, 표현력, 성능 등에 대한 검증이 필요하다.

참조 문헌

[1] RDF, Resource Description Framework (RDF): Concepts and Abstract Syntax. Graham Klyne and Jeremy Carroll. W3C Working Draft, November 2002.  
 [2] RDF Schema, RDF Vocabulary Description Language 1.0: RDF Schema. Dan Brickley and R. V. Guha. W3C

Working Draft 23 January 2003.

[3] OWL Guide, OWL Web Ontology Language Guide. Mike Smith, Chris Welty, and Deborah L. McGuinness. W3C Candidate Recommendation 18 August 2003.  
 [4] OWL Abstract Syntax and Semantics, OWL Web Ontology Language Semantics and Abstract Syntax, Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks. W3C Candidate Recommendation 18 August 2003.  
 [5] OWL Reference, OWL Web Ontology Language Reference. Mike Dean, Guus Schreiber, Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. W3C Candidate Recommendation 18 August 2003.  
 [6] OWL Overview, OWL Web Ontology Language Overview. Deborah L. McGuinness and Frank van Harmelen. W3C Candidate Recommendation 18 August 2003.  
 [7] OWL Requirements, OWL Web Ontology Language Use Cases and Requirements. Jeff Heflin. W3C Candidate Recommendation 18 August 2003.  
 [8] DAML-OIL, DARPA Agent Markup Language-Ontology Inference Layer, <http://www.daml.org/>