

# OWL 온톨로지 합병도구의 설계 및 구현

김재홍<sup>0</sup> 손주찬

한국전자통신연구원 인터넷컴퓨팅연구부 비즈니스지식처리연구팀  
(jhkim504<sup>0</sup>, jcsohn)<sup>0</sup>@etri.re.kr

## A Design and Implementation of Ontology Merging Tool Supporting OWL

Jaehong Kim<sup>0</sup> Joo-Chan Sohn

Business Knowledge Research Team, Internet Computing Department, ETRI

### 요 약

최근 시맨틱 웹의 중요성이 부각되면서 다양한 분야에서 이와 관련된 연구가 활발히 진행되고 있다. 시맨틱 웹과 관련된 연구 중 중요한 부분이 온톨로지와 규칙 및 추론이며, 이들 연구가 실용적인 측면에서 빠르게 진행되기 위해서는 연구를 위해 사용가능한 온톨로지가 많이 구축되어 있어야 한다. 그러나, 온톨로지의 구축은 많은 노력과 시간을 요하기 때문에 온톨로지 합병을 통한 기존 온톨로지의 재사용이 반드시 필요하다. 이미 온톨로지 합병을 위한 도구가 다수 존재하나 이들이 시맨틱 웹에서의 온톨로지 언어인 OWL을 직접적으로 지원하지 않는다는 단점이 있다. 본 논문에서는 기존 온톨로지 합병도구의 분석을 통해 OWL 온톨로지 합병도구 개발을 위해 필요한 기본적인 기능들을 추출하고, 이를 설계 및 구현하였다. 기존 온톨로지를 OWL로 변환해 주는 도구와 개발된 도구를 사용하여 현재 다수 존재하는 기존 온톨로지를 재활용함으로써 시맨틱 웹의 활용 촉진에 기여할 수 있을 것으로 기대된다.

### 1. 서론

시맨틱 웹은 웹 상의 정보에 잘 정의된 의미를 부여함으로써 사람뿐만 아니라 컴퓨터도 쉽게 문서의 의미를 해석할 수 있도록 하여 컴퓨터를 이용한 정보의 검색 및 해석 통합 등의 업무를 자동화하기 위한 목적으로 제안되었으며, 그 계층구조는 그림1과 같다[1].

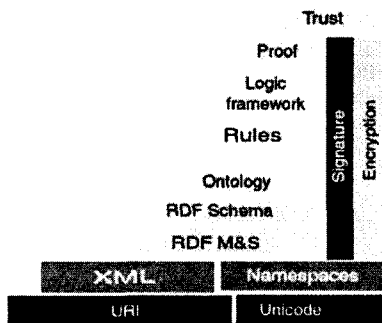


그림 1 시맨틱 웹의 구조

그림1의 계층구조 중 온톨로지(Ontology)는 특정 도메인의 지식을 명시적으로 표현하기 위한 개념들, 개념 사이의 관계, 개념의 속성 및 특성, 속성 및 특성에 부여된 제약조건 및 객체들로 표현된다. 이 온톨로지를 통하여 특정 도메인의 단어를 공통으로 정의하고, 지식을 공유할 수 있다. 규칙(Rules)과 논리계층(Logic framework)은 이러한 온톨로지에 기반하여 질의와 추론을 가능하게 한다[2].

최근 시맨틱 웹의 중요성이 부각되면서 다양한 분야에서

이와 관련된 연구가 활발히 진행되고 있으나, 실용적인 측면에서의 연구는 그리 많은 편이 아니다. 이것은 시맨틱 웹 활용의 효용성을 보여 줄 수 있는 어플리케이션의 부족에 기인하며, 결국 어플리케이션을 개발할 수 있는 온톨로지가 부족하다는 것을 의미한다. 따라서, 온톨로지를 쉽게 생성할 수 있는 도구 특히 기존의 온톨로지를 재활용할 수 있도록 도와주는 도구가 필요하다.

온톨로지 합병을 위한 도구는 이미 다수 존재하나[3], 이들이 현재까지는 시맨틱 웹에서의 온톨로지 언어인 OWL을 직접적으로 지원하지 못하고 있다는 단점이 있다.

본 논문에서는 기존 온톨로지 합병도구의 분석을 통해 OWL 온톨로지 합병도구 개발을 위해 필요한 기본적인 기능들을 추출하고, 이를 설계 및 구현하였다. 기존 온톨로지를 OWL로 변환해 주는 도구와 개발된 도구를 함께 사용하면 현재 다수 존재하는 기존 온톨로지를 재활용함으로써 시맨틱 웹의 활용 촉진에 기여할 수 있을 것으로 기대된다.

### 2. 온톨로지 합병을 위해 기본적으로 필요한 기능

온톨로지 합병도구의 개발을 위해 기본적으로 필요한 기능은 다수의 온톨로지를 합병할 때 발생하는 충돌을 찾아 그 원인을 해결하는 것으로 아래 각 절에서 세부적으로 설명한다.

#### 2.1 충돌의 종류

가장 기본적인 충돌의 종류는 이름 충돌이며, 동일한 이름을 가진 개체가 두개 존재할 경우 이에 대한 해결이 필요하다. 추가적으로 합병된 온톨로지에서 존재하지 않는 개체에 대한 참조가 있을 경우에도 문제를 해결해야 한다.

2.2 합병 대상

기본적인 합병의 대상은 클래스와 속성, 그리고 클래스의 인스턴스(Individual)이다.

2.3 기본 연산

합병된 온톨로지는 이미 존재하는 온톨로지를 재활용하는 것이므로 복사기능이 필요하여, 합병된 온톨로지에서 충돌을 해결하기 위해서는 합병과 삭제 연산이 필요하다.

이하의 설명에서 다음과 같은 기호를 설명의 편의를 위해 사용한다.

- $O_1$ : 소스 온톨로지 1,  $O_2$ : 소스 온톨로지 2,  $O_m$ : 합병된 온톨로지,  $O_m$ :  $O_m$  온톨로지에서 Merge, Add, Remove 연산을 수행한 후의 온톨로지
- $C_1$ :  $O_1$ 의 클래스,  $C_2$ :  $O_2$ 의 클래스,  $C_m$ :  $O_m$ 의 클래스
- $C_{m1}, C_{m2}$ :  $O_m$ 으로 이동한 후의  $C_1, C_2$
- $P_1$ :  $O_1$ 의 속성,  $P_2$ :  $O_2$ 의 속성,  $P_m$ :  $O_m$ 의 속성
- $P_{m1}, P_{m2}$ :  $O_m$ 으로 이동한 후의  $P_1, P_2$
- $I_1$ :  $O_1$ 의 인스턴스,  $I_2$ :  $O_2$ 의 인스턴스,  $I_m$ :  $O_m$ 의 인스턴스
- $I_{m1}, I_{m2}$ :  $O_m$ 으로 이동한 후의  $I_1, I_2$

기본적으로 필요한 연산은 다음과 같이 클래스, 속성, 인스턴스에 대한 합병, 복사, 삭제 연산이다.

- 클래스의 복사  $O_m = Add(O_m, C_1 | C_2)$
- 속성의 복사  $O_m = Add(O_m, P_1 | P_2)$
- 인스턴스(Individual)의 복사  $O_m = Add(O_m, I_1 | I_2)$
- 클래스의 합병  $C_m = Merge(C_{m1}, C_{m2})$
- 속성의 합병  $P_m = Merge(P_{m1}, P_{m2})$
- 인스턴스(Individual)의 합병  $I_m = Merge(I_{m1}, I_{m2})$
- 클래스의 삭제  $O_m = Remove(O_m, C_m)$
- 속성의 삭제  $O_m = Remove(O_m, P_m)$
- 인스턴스(Individual)의 삭제  $O_m = Remove(O_m, I_m)$

3. 기본 연산의 구현시 고려사항

클래스, 속성 및 인스턴스의 합병, 복사, 삭제시에는 다음과 같은 사항에 대한 고려가 필요하다.

3.1 복사시 고려사항

- 클래스 복사
  - 상위 클래스( $C_{sup}$ )를 가지는 경우:  $C_{sup}$ 를 가지는데,  $C_{sup}$ 가  $O_m$ 에 존재하지 않는 경우  $C_{sup}$ 를 함께 복사 독립된 클래스로 복사
  - 하위 클래스( $C_{sub}$ )를 가지는 경우: 클래스 자체만 복사 또는  $C_{sub}$ 를 함께 복사
  - 인스턴스를 가지는 경우: 인스턴스를 함께 복사 또는 클래스만 복사
- 속성(P)을 가지는 경우: P를 함께 복사 또는 클래스만 복사, P를 함께 복사할 경우 P의 정의가  $O_m$ 에 없을 때  $O_1$ (또는  $O_2$ )에 있는 P의 정의를  $O_m$ 에 복사
- 속성 복사
  - 속성 복사시: 속성을 복사하는데, 참조되는 클래스 정보가  $O_m$ 에 없을 경우 속성 복사를 취소하거나 클래스를 함께 복사
  - 상위 속성( $P_{sup}$ )을 가지는 경우:  $P_{sup}$ 를 가지는데,  $P_{sup}$ 가  $O_m$ 에 존재하지 않는 경우  $P_{sup}$ 를 함께 복사 또는 독립된

속성으로 복사

- 하위 속성( $P_{sub}$ )을 가지는 경우: 속성 자체만 복사 또는  $P_{sub}$ 를 함께 복사

3.2 합병시 고려사항

- 클래스 합병
  - 상위 클래스를 가지는 경우:  $C_{m1}, C_{m2}$ 의 상위 클래스  $C_{sup-m1}, C_{sup-m2}$ 를  $C_m$ 의 상위 클래스로 한다.
  - 하위 클래스를 가지는 경우:  $C_{m1}, C_{m2}$ 의 하위 클래스  $C_{sub-m1}, C_{sub-m2}$ 를  $C_m$ 의 하위 클래스로 한다.
  - 인스턴스를 가지는 경우:  $C_{m1}$ 의 인스턴스와  $C_{m2}$ 의 인스턴스를  $C_m$ 의 인스턴스로 한다.
  - 속성을 가지는 경우:  $C_{m1}$ 의 속성과  $C_{m2}$ 의 속성을  $C_m$ 의 속성으로 하고 각 속성의 도메인을  $C_m$ 으로 한다.
- 속성 합병
  - 상위 속성을 가지는 경우:  $P_{m1}, P_{m2}$ 의 상위 속성  $P_{sup-m1}, P_{sup-m2}$ 를  $P_m$ 의 상위 속성으로 한다.
  - 하위 속성을 가지는 경우:  $P_{m1}, P_{m2}$ 의 하위 속성  $P_{sub-m1}, P_{sub-m2}$ 를  $P_m$ 의 하위 속성으로 한다.

3.3 삭제시 고려사항

- 클래스 삭제
  - 하위 클래스( $C_{sub}$ )를 가지는 경우:  $C_{sub}$ 를 함께 삭제 또는 존속, 존속시 상위 클래스  $C_{sup}$ 가 존재하면  $C_{sub}$ 를  $C_{sup}$ 의 하위 클래스로 하고, 그렇지 않으면  $C_{sub}$ 단독으로 존재
  - 인스턴스를 가지는 경우: 함께 삭제
  - 속성을 가지는 경우: 클래스에의 종속성은 삭제, 속성 정의 자체는 참조되는 곳이 있으면 남겨두고, 참조되는 곳이 없을 때는 옵션으로 처리
- 속성(정의) 삭제
  - 속성 삭제시: 이 속성을 참조하는 곳이 있으면 삭제 불가
  - 하위 속성( $P_{sub}$ )을 가지는 경우:  $P_{sub}$ 를 함께 삭제 또는 존속, 존속시  $P_{sup}$ 가 존재하면  $P_{sub}$ 를  $P_{sup}$ 의 하위 속성으로 하고, 그렇지 않으면  $P_{sub}$ 단독으로 존재

4. 온톨로지 합병도구의 구현

4.1 도구의 처리 흐름

본 논문에서 구현한 온톨로지 합병 도구의 처리 흐름은 그림 2와 같다.

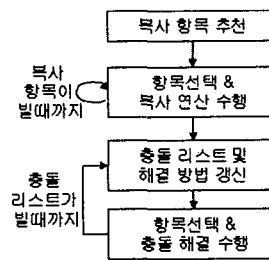


그림 2 온톨로지 합병도구의 처리 흐름

합병을 할 소스 온톨로지 2개( $O_1, O_2$ )를 오픈하면 클래스, 속성, 인스턴스의 복사할 대상 리스트가 만들어진다. 이 대상 중에서 합병된 온톨로지( $O_m$ )에 복사할 항목을 선택하여 복사를 수행한다. 이 과정은 원하는 것을 모두 복사할 때까지

계속된다.  $O_m$ 이 생성되어  $O_1, O_2$ 의 클래스, 속성, 인스턴스가 추가되는 과정에 충돌리스트가 항상 갱신된다. 충돌리스트는 이름 충돌과 없는 개체에 대한 참조로 구성되며, 충돌 내용과 함께 이것을 해결하는 방법(합병, 삭제, 없는 참조 추가)이 제시된다. 이때 해결을 선택하면 충돌을 해결하기 위한 연산이 수행된다. 원하는 항목을 모두 복사한 상태에서 충돌이 하나도 남지 않은 상태이면 온톨로지의 합병 과정이 완료된 상태이다.

#### 4.2 도구의 구현

온톨로지 합병 도구는 JDK 1.4.1을 사용하여 Java 어플리케이션으로 구현하였으며, OWL 온톨로지 파서로 Jena 2.0 Beta 1[5]을 이용하였다.

구현된 온톨로지 합병도구는 사용자의 편의를 위해 3가지 합병 방식을 제공하며, 사용자가 자신의 재활용 방식에 따라 선택하여 사용할 수 있다.

- 합병 후 해결:  $O_1$ 과  $O_2$ 의 모든 개체를  $O_m$ 으로 복사한 후 충돌을 제거해 나가는 방식
- 지정 온톨로지 우선:  $O_1$ 의 모든 개체를  $O_m$ 으로 복사한 상태에서  $O_2$ 의 개체들을  $O_m$ 에 추가하면서 충돌을 해결해 나가는 방식
- 대화형:  $O_m$ 이 빈 상태에서  $O_1$ 과  $O_2$ 의 개체들 중 재활용할 것을  $O_m$ 에 하나씩 추가하면서 충돌을 해결해 나가는 과정

그림 3은 구현된 온톨로지 합병도구의 실행 화면이다.

#### 5. 결론 및 향후 연구

본 논문에서는 기존 온톨로지 합병도구의 분석을 통해 OWL 온톨로지 합병도구 개발을 위해 필요한 기본적인 기능들을 추출하고, 이를 설계 및 구현하였다. 기존 온톨로지를 OWL로 변환해 주는 도구와 개발된 도구를 사용하여 현재 다수 존재하는 기존 온톨로지를 재활용함으로써 시맨틱 웹의 활용 촉진에 기여할 수 있을 것으로 기대된다.

현재의 도구는 가장 기본적인 기능들만 제공하는 상태이며, 언어정보와 OWL 자체의 특징을 응용하여 합병도구의 활용성을 높이기 위한 연구를 하고 있으며, 온톨로지 정렬 기능도 제공할 예정이다. 또한, 온톨로지 작성 및 편집 작업과의 효율적인 연계를 위해 OWL 온톨로지 저작도구와도 통합할 예정이다.

#### 참고문헌

- [1] T. Berners-Lee, "The Semantic Web," <http://www.w3.org/2002/Talks/04-sweb/>, Academic disussion talk, Japan Prize 2002.
- [2] 조성정, 김진형, "시맨틱 웹의 응용 사례 연구," 정보과학회지 제21권, 제3호, pp11-17, 2003.
- [3] OntoWeb Consortium, "Deliverable 1.3: A survey on ontology tools", [http://www.ontoweb.org/download/deliverables/D13\\_v1-0.zip](http://www.ontoweb.org/download/deliverables/D13_v1-0.zip), pp35-51, 2002.
- [4] Noy, N.F. and Musen, M.A., "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment," 17<sup>th</sup> National Conference on Artificial Intelligence(AAAI-2000), Austin, TX, 2000.
- [5] Jena 2.0, <http://www.hpl.hp.com/semweb/jena2.htm>.

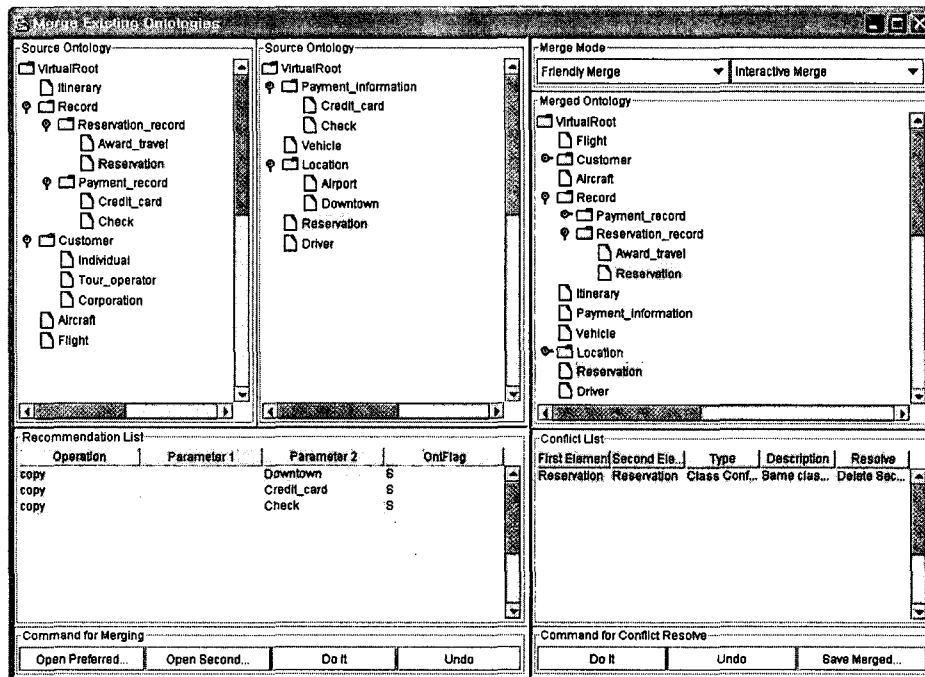


그림 3 온톨로지 합병도구 수행 화면