

사용자 요구사항의 Annotation을 위한 OWL 기반 프로토타입 구현

김기훈*, 조성훈*, 이무훈*, 장창복*, 조현규**, 송병열**, 최의인*

*한남대학교 컴퓨터공학과

**한국전자통신연구원

e-mail:khkim@dblab.hannam.ac.kr

The Implementation of the Prototype based on OWL for the Annotating User Request

Ki-Hoon Kim*, Sung-Hoon Cho*, Moo-Hun Lee* Chang-Bok
Jang*, Hyun-Kyu Cho**, Byoung-Youl Song**, Eui-In Choi*

*Dept of Computer Engineering, Han-Nam University

**Electronics and Telecommunications Research Institute

요 약

본 논문에서는 시맨틱 웹의 핵심 기술인 웹 온톨로지 언어(OWL)를 이용하여 개념에 대한 의미와 개념간의 관계를 지식으로 표현한 온톨로지를 기반으로 사용자가 요구하는 웹 서비스에 대한 요구사항을 수집할 수 있는 인터페이스를 제공하고 사용자의 요구사항을 기계가독형 OWL문서로 annotation할 수 있는 프로토타입을 구현하였다.

1. 서론

오늘날의 웹은 개인적인 정보 교류의 공간이면서 기업 비즈니스 활동의 주무대이다. 또한 정보 저장소로서 쉽게 접근할 수 있는 장점을 제공한다.

이러한 현재의 웹은 웹 브라우저를 기반으로 URL을 이용하거나 검색엔진을 통해 서비스를 이용하는 방식을 취하고 있으며 웹상의 정보를 표현하고 서비스를 기술하는 방법은 표현 위주의 마크업 언어(HTML)를 사용하여 인간 중심적인 콘텐츠로 구성되어 있다. 이러한 특성은 사용자가 정보를 검색하고 서비스를 이용하는데 사용자의 개입을 반드시 필요로 하며, 사용자가 요구하는 정보를 찾기까지 사용자가 많은 노력을 감수해야만 한다. 또한 근래 SOAP, WSDL, UDDI 등과 같은 XML을 기반 표준

을 적용한 웹 서비스 기술이 e-비즈니스를 위한 기술로 주목을 받고 있다. 웹 서비스 기술은 기존의 분산 컴퓨팅 기술(CORBA, DCOM, RMI 등)에 비해 단순하고 확장이 용이하며, 이기종 시스템간의 상호운용을 가능하게 한다. 기업의 입장에서 웹을 통해 비즈니스를 하는 것은 많은 이점을 제공하지만 표준 기술들이 구현과 실행 위주의 기술(description)에 치중되어 있어서 여전히 인간의 개입을 통해 서비스를 검색하고 서비스를 이용하는 기본적인 과정을 거치게 된다. 최근에 이러한 문제점을 해결할 수 있는 개념으로 시맨틱 웹이 제안되었다.

시맨틱 웹은 현재의 웹과 같이 단순 웹 자원에 대한 링크를 통해 정보를 제공하지 않고 의미적 연결을 통해 정보를 제공한다. 즉 정보의 의미를 개념으로 정의하고 개념 간의 관계성을 표현함으로써 정보를 공유하게 된다. 따라서 웹상의 정보를 수집하고 처리하기 위해 인간의 전적인 개입이 요구되지 않으며 자동화된 에이전트를 통해 정보의 의미와 정

본 연구는 한국전자통신연구원의 "사용자 요구 기반 웹 서비스 프로우 생성기술 연구"의 연구 결과임

보간의 관계성이 파악되고 이를 통해 정확한 정보의 검색과 추론을 통한 새로운 지식의 생성이 가능하게 된다[2, 4].

그래서 앞서 기술한 웹 서비스가 웹 어플리케이션간의 통합과 상호운용을 가능하게 하지만 인간의 개입을 반드시 필요하다는 문제점을 보완하기위해서 의미적 처리가 가능한 시맨틱 웹 기술을 웹 서비스 기술에 적용하여 자동적인 웹 서비스의 발견, 실행, 조합 등을 가능하게 하는 시맨틱 웹 서비스에 관련된 연구가 진행되고 있다[2, 7, 8].

웹 서비스의 자동화를 위해서는 우선 사용자가 요구하는 서비스가 무엇인지 분석할 수 있어야 한다. 즉 사용자가 원하는 서비스가 정확히 무엇인지 구별하고 그 의미가 결정되어야 한다는 것이다.

이와 같이 사용자 요구사항에 대한 의미를 파악하고 구분하는 것은 웹 온톨로지와 추론기술을 이용하여 처리할 수 있다. 그러나 아직까지 개념적인 논의에 지나지 않아 가시적인 연구결과의 제시가 요구되고 있는 상황이다. 따라서 본 논문에서는 OWL 기술된 웹 서비스 온톨로지를 이용하여 다양한 사용자의 요구사항을 파악할 수 있는 인터페이스를 제공하고 기계에 의해 의미적으로 해석될 수 있는 OWL 문서로 사용자의 요구사항을 annotation할 수 있는 프로토타입을 구현하려고 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로서 온톨로지에 대한 개념과 대표적인 웹 온톨로지 언어인 OWL을 간략히 기술하고 XML을 기반으로 사용자 요구사항을 수집하는 WSOM에 대해 기술한다. 3장에서는 본 논문에서 사용자의 요구사항을 파악하여 OWL 기반으로 annotation하는 Preference Annotator에 대해 설명하고 마지막으로 4장에서 결론 및 향후 연구과제를 기술한다.

2. 관련연구

2.1 온톨로지

온톨로지(ontology)에 대한 정의는 분야에 따라 다르게 정의 될 수 있지만 Gruber가 내린 온톨로지의 정의는 다음과 같다. "An ontology is a formal, explicit specification of a shared conceptualization of a domain of interest"[6] 위 문장을 풀어서 해석하면 형식적(formal)이란 기계가 판독할 수 있는 형태로 표현한다는 것을 의미하고 명시적(explicit)이라는 것은 개념들의 종류와 개념들간에 규칙을 명확하

게 정의 한다는 것을 의미한다. 그리고 공유된(shard) 이라는 말은 특정영역에 관계자들이 동의하여 합의된 정의라는 것을 의미하고 개념화(conceptualization)라는 말은 관련 있는 개념들을 식별하여 그 현상을 추상화한 모델로 설명하는 것이다 [5]. 즉 온톨로지는 특정영역에 관련자의 동의에 의해 개념과 개념들간의 관계와 규칙을 형식적이고 명확하게 기술한 명세서라고 기술할 수 있다.

2.2 OWL

OWL은 W3C가 주도해온 시맨틱 웹 형성의 기반이 되는 온톨로지를 위한 언어로서 인간중심의 표현이기 보다는 기계 중심 표현으로써 기계에 의해서 판독되고 처리가 용이하도록 설계되었다. 이 OWL은 XML, RDF, RDF schema 보다 개념과 개념간의 관계 등을 표현하는데 있어 더 용이하다. 그리고 OWL은 DAML+OIL의 개선된 형태이며 세가지 향상된 표현을 제공하는 서브언어(sublanguage)인 OWL Lite, OWL DL, OWL Full로 구성되어 있다 [1].

2.3 WSOM

WSOM(Web Services Outsourcing Manger)은 IBM의 alphaworks 프로젝트에 결과로서 사용자 요구사항에 기반을 둔 웹 서비스 플로우의 동적인 조합이 가능한 프레임워크이다. WSOM에 의해 파악된 사용자의 요구사항은 비즈니스 프로세스 아웃소싱을 위한 annotation 문서를 생성하기 위해 분석되고 최적화된다[3].

이러한 WSOM은 사용자 요구사항에 기반을 둔 웹 서비스의 검색이 가능한 도구이다. 그러나 사용자가 요구하는 서비스에 대해 제한적인 요구조건만을 제시하고 있기 때문에 웹상에서 제공되는 다양한 서비스를 수용하기에는 어려움이 있다. 또한 XML을 기반으로 annotation되기 때문에 기계에 의한 의미적 처리가 불가능하다.

3. 시스템 설계 및 구현

본 논문에서는 OWL로 기술된 웹 서비스 온톨로지를 이용하여 다양한 서비스에 대한 사용자의 요구사항을 제시할 수 있는 인터페이스와 기계에 의해 의미적으로 해석될 수 있도록 OWL 문서로 사용자의 요구사항을 annotation하는 Preference Annotator

를 구현하였다.

3.1 개발환경

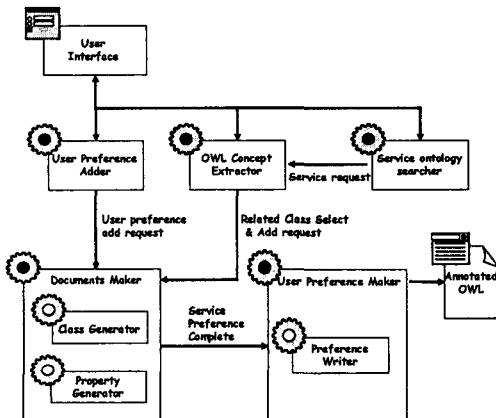
Preference Annotator의 개발을 위한 라이브러리로 Hewlett-Pakckard사의 Jena 2.1 시맨틱 웹 툴킷을 이용하였다.

Jena 2.1은 시맨틱 웹을 위한 Java Framework으로서 RDF, RDFS, N-TRIPLE, DAML, DAML+OIL, OWL을 지원한다. Jena 2.1에서는 OWL을 생성 및 read하기 위해 OntModel 클래스를 제공한다. OntModel 클래스는 OWL의 구문을 저장하기 위해 Graph에 기반한 자료구조를 가진다. Jena는 OWL을 RDF(Resource Description Framework)의 확장으로 보고 OWL 구문을 생성하기 위해 RDF의 Statement를 사용한다. 이 Statement는 N-TRIPLE(Subject, Predicate, Object)의 표현과 정확히 매칭되기 때문에 RDF 문서 구성이 용이하다.

3.2 Preference Annotator의 구성

우선 Preference Annotator의 구성 모듈을 설명하기 전에 [그림 1]에 포함되지 않은 ontology list를 간략히 언급하고 Preference Annotator의 구성을 설명하고자 한다. ontology list는 OWL로 기술된 온톨로지를 관리하기 위해 온톨로지에 대한 메타데이터(OWL URI, 파일명, 키워드, OWL에 대한 description, OWL의 분류정보)를 XML로 기술한 파일이다. 이러한 ontology list는 Preference Annotator와 독립적인 서버환경에서 관리되며 Preference Annotator의 요청에 의해 호출된다.

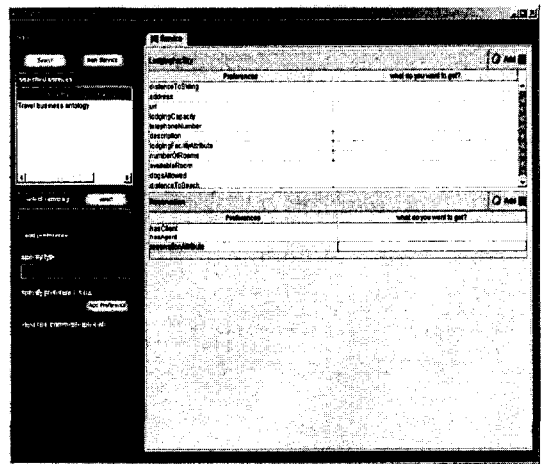
Preference Annotator의 구성은 [그림 1]과 같다.



[그림 1] Preference Annotator 구성 모듈

• User Interface

User Interface는 사용자와 Preference Annotator가 상호작용하기 위한 모듈로 [그림 2]와 같이 구성되어 있다. User Interface에서 ①은 ontology list에 검색결과로 OWL에 대한 description 정보의 리스트를 보여주고, ②는 ①에서 선택한 description의 연관된 OWL에서 추출된 Class 리스트를 보여준다. 그리고 ③은 사용자가 원하는 요구사항이 OWL 문서에 없을 경우 새로운 항목을 추가할 수 있는 부분이며 ④와 ⑤는 서비스에 대한 사용자의 요구사항과 요구사항에 대한 사용자의 입력 데이터의 쌍으로 구성되어 있다.



[그림 2] Preference Annotator UI

• Service Ontology Searcher

사용자가 입력한 키워드를 통해 온톨로지 정보가 기술된 ontology list의 keyword 요소의 값과 비교하여 일치하는 description 정보들을 반환한다.

• OWL Concept Extractor

사용자가 자신이 원하는 서비스를 Service Ontology Searcher를 통해 검색하고 검색결과 리스트 중 사용자가 선택한 특정 OWL 문서에서 사용자의 요구사항을 추출하기 위해 OWL의 Class 구문을 추출하여 반환한다.

• Document Maker

사용자가 OWL Concept Extractor에 의해 추출된 Class를 선택하면 사용자의 요구사항을 가진 Class로 추가하는 일을 한다. 선택된 Class는 Class의 LocalName을 통해 원본 OWL 문서상에

서 다시 한번 검색하여 owl:Class와 owl:Restriction, rdfs:subClassOf에 대한 정보를 추출한다.

· Class Generator

Document Maker에서 전달받은 owl:Class, owl:Restriction, rdfs:subClassOf를 통해 사용자 OWL 문서를 생성한다. 그러나 owl:Restriction이 있을 경우에는 대응되는 Property가 존재하기 때문에 OWL 문서상에 먼저 Property를 생성해야만 한다. 이를 위해서 Property Generator가 호출되어 각각의 Property를 생성 추가한다. 만약 owl:Restriction이 존재하지 않을 경우에는 Property Generator를 호출하지 않는다.

· Property Generator

Class Generator에서 생성할 owl:Class에 owl:Restriction이 존재할 경우에 호출되며, 각 Property를 원본 OWL 문서에서 검사하여 유형에 대한 정보를 획득한 뒤에 유형 정보에 따라 owl:DatatypeProperty, owl:ObjectProperty, owl:SymmetricProperty, owl:TransitiveProperty, owl:FunctionalProperty 등으로 사용자 OWL 문서에 생성 추가한다.

· User Preference Adder

사용자가 원하는 요구사항이 Class에 없을 경우 직접 추가할 수 있는 기능을 수행한다.

· User Preference Maker

사용자가 요구사항으로 선택한 owl:Class의 Property 중에서 데이터가 있는 Property만을 선택적으로 저장하는 기능을 수행한다.

본 연구를 통해 개발된 Preference Annotator는 완성도 있는 어플리케이션의 구현하기 보다는 OWL로 기술된 온톨로지를 이용하여 사용자의 요구사항을 수집하고 기계가독형 언어로 annotation하는 데 초점을 두고 연구되었다.

4. 결론 및 향후 연구

본 논문에 프로토타입으로 구현한 Preference Annotator로 annotation된 정보는 추론기술을 이용하여 어플리케이션에 의한 분석을 가능하게 하며 이를 통해 사용자가 요구하는 최적의 서비스를 검색하는 것이 가능하게 된다. 그러나 서론에서 간략히 언급한 시맨틱 웹의 현실화와 본 논문에서 제시한 다

양한 서비스에 대한 사용자의 요구정보를 수집하기 위해서는 웹 온톨로지에 대한 인프라가 구축되어야 하는 선결 과제가 남아있으며, 향후 연구로 annotation된 정보를 이용하여 사용자의 선호도 (preference)를 활용하는 방법에 대한 연구가 필요하다.

참고문헌

- [1] Deborah L. McGuinness and Frank van Harmelen "OWL Web Ontology language Overview", 2004, <http://www.w3.org/TR/2004/REC-owl-features-20040210/>.
- [2] Abhijit Patil, Swapna Oundhakar, Amit Sheth, Kunal Verma "METEOR-S Web service Annotation Framework" LSDIS Lab, Dept of CS, University of Georgia.
- [3] IBM, <http://alphaworks.ibm.com/tech/WSOM>
- [4] 하영국 "Information Retrieval System for the Semantic Web", CS774 Term Project Report.
- [5] 양정진 "시맨틱 웹에서의 온톨로지 공학", 한국정보과학회, 제21권 제 3호, P28-35, 2003
- [6] Gruber, T., "A translation approach to portable ontologies", Knowledge Acquisition, Vol. 5, No. 2, pp.199-220, 1993.
- [7] David Martin, "OWL-S 1.0 Release", 2004, <http://www.daml.org/services/owl-s/1.1B/owl-s/owl-s.html>.
- [8] <http://www.w3.org/2002/ws/>