

UML 및 OCL을 이용한 서비스 온톨로지 설계 방안에 관한 연구

이윤수*, 정인정*

*고려대학교 전산학과

e-mail: {arzhna, chung}@korea.ac.kr

Study Ontology Design Scheme Using UML and OCL

Yun-Su Lee*, In-Joeng Chung*

*Dept of Computer Science, Korea University

요 약

현재 지능형 웹 서비스를 위한 서비스 온톨로지의 생성은 서비스 개발자의 휴리스틱에 의존하여 많은 시간과 비용을 소모할 뿐만 아니라 서비스와 서비스 온톨로지간의 완벽한 매핑이 어렵다는 문제점을 가지고 있다. 또한 서비스 온톨로지를 기술하기 위한 마크업 언어를 서비스 개발자가 단기간 내에 학습하기에 많은 어려움이 있는 실정이다.

본 논문에서는 지능형 웹 서비스를 위한 서비스 온톨로지 생성의 문제점들을 해결하기 위해 MDA 방법론을 사용하여 서비스 온톨로지를 효율적으로 설계 및 생성하기 위한 방안을 제안한다. 본 논문에서 제안하는 방안은 MDA를 기반으로 널리 사용되고 있는 UML을 사용하여 웹 서비스 모델을 설계하고 이를 OWL-S에 최적화된 모델로 변환한 후 XMI를 통해 OWL-S로 기술되는 서비스 온톨로지로 변환한다. 본 논문에서 제안하는 방안은 이미 널리 사용되는 UML과 같은 소프트웨어 공학적 방법을 사용하기 때문에 서비스 개발자들이 쉽게 서비스 온톨로지를 구축할 수 있으며 하나의 모델로부터 서비스와 서비스 온톨로지 모델을 동시에 이끌어 낼 수 있는 장점을 가진다. 또한 모델로부터 자동적으로 서비스 온톨로지를 생성함으로써 시간과 비용을 절감할 수 있는 효과를 얻을 수 있다. 그리고 플랫폼 변화와 같은 외부 환경 변화에 유연하게 대처할 수 있다. 끝으로 본 논문에서는 제안된 방안의 타당성을 검증하기 위해 실제로 웹 서비스 모델을 설계하고 서비스 온톨로지를 생성하는 예를 보인다. 또한 생성된 서비스 온톨로지가 올바르게 생성되었는지를 유효성 검사를 통해 검증한다.

1. 서론

지능형 웹 서비스는 시맨틱 웹과 에이전트 기술을 통하여 자동적인 서비스의 발견, 호출, 결합 및 상호운영, 실행감시 및 복구를 수행하는 것을 목적으로 제안되었다[1]. 이러한 기능의 실현을 위해서는 컴퓨터가 지식을 추론하고 처리할 수 있게 하기 위한 온톨로지가 필수적으로 요구된다.

그러나 현재 지능형 웹 서비스를 위한 서비스 온톨로지는 서비스 구축과는 별개로 서비스 개발자의 휴리스틱에 의존하여 생성되고 있으며 이로 인한 많은 시간과 비용이 소모된다는 단점을 가지고 있다. 또한 서비스와 서비스 온톨로지가 별개로 구축되어 완벽한 매핑이 어려울 뿐만 아니라 서비스 온톨로지 기술을 위한 마크업 언어는 서비스 개발자가 단기간 내에 학습하기에 많은 어려움이 있는 실정이다[2].

따라서 우리는 이러한 문제점들을 해결하기 위해 MDA (Model Driven Architecture)[1] 방법론을 사용하여 서비스 온톨로지를 효율적으로 설계 및 생성하기 위한 방법을

제안한다. 우리가 제안하는 방안은 MDA를 기반으로 UML[2]을 사용하여 웹 서비스 모델을 설계하고 구축하는 과정에서 생성되는 모델을 재사용한다. 즉, 플랫폼 독립적인 웹 서비스 모델을 서비스 온톨로지 기술 언어인 OWL-S[3]에 종속적인 모델로 변환한 후 XMI[4]를 통해 OWL-S 서비스 온톨로지로 변환한다.

우리가 제안하는 방안은 이미 널리 사용되는 UML과 같은 소프트웨어 공학적 방법을 사용하기 때문에 서비스 개발자들이 쉽게 서비스 온톨로지를 구축할 수 있으며 하나의 모델로부터 서비스와 서비스 온톨로지 모델을 동시에 이끌어 낼 수 있는 장점을 가진다. 또한 모델로부터 자동적으로 서비스 온톨로지를 생성함으로써 시간과 비용을 절감할 수 있다. 그리고 플랫폼 변화와 같은 외부 환경 변화에 유연하게 대처할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 사용되는 기술들에 대해 간략히 소개하며 MDA를 사용하여 온톨로지를 설계 및 생성하는 연구들에 대해 소개한다. 그리고 3장에서는 본 논문에서 제안하는 방안에 대해 기

술하며 4장에서는 제안된 방안을 검증하기 위한 실험 결과를 기술한다. 끝으로 5장에서는 결론을 맺고 향후과제에 대해 기술한다.

2. 기반 개념 및 관련 연구

2.1 MDA

MDA는 모델을 기반으로 소프트웨어를 구축하기 위해 OMG에 의해 제안된 소프트웨어 공학의 표준 패러다임이다[1]. 이 접근방법은 모델링 언어 및 구현 언어들의 메타모델간의 매핑을 통하여 상호 변환이 가능하도록 지원함으로써 거대한 소프트웨어 애플리케이션을 효율적으로 개발하기 위해 사용된다. 즉, 플랫폼에 독립적인 모델링 언어로 모델을 설계하고 이를 특정 플랫폼에 기반한 구현에 가까운 모델로 변환하여 보다 쉽게 애플리케이션을 개발할 수 있는 능력을 제공한다. MDA를 위한 OMG의 표준은 UML, MOF[10], XMI등이 있다.

2.2 OWL-S

웹 서비스를 위한 서비스 온톨로지를 기술하기 위해 OWL에서 파생된 서비스 온톨로지 마크업 언어이다. OWL-S는 어떠한 서비스를 제공하는지를 기술하는 서비스 프로파일, 서비스의 행동을 기술하는 서비스 모델, 서비스에 접근하기 위한 방법을 기술하는 그라운드링으로 구성되어 있다. (그림 1)은 OWL-S의 상위 온톨로지 구조를 나타낸다[3].



(그림 1) OWL-S의 구성요소

2.3 관련 연구

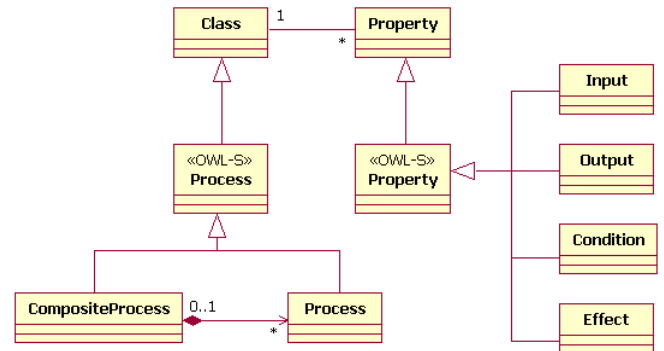
UML을 이용하여 도메인 온톨로지를 생성하기 위한 연구는 많이 진행되었지만 서비스 온톨로지를 생성하기 위한 연구는 별로 활발하지 않았다. 서비스 온톨로지는 서비스의 행동을 기술하기 위한 온톨로지이기 때문에 기존에 시도되었던 UML의 클래스 다이어그램을 통한 방법만으로는 충분하지 않다. [13]에서는 UML의 상태도(State Chart)를 이용하여 서비스를 모델링하는 방안을 제안하고 있지만 이는 여러 개의 복합 프로세스(Composite Process)들로 구성된 복잡한 서비스를 나타내기에는 한계가 있다[12]. 이러한 문제점을 언급하고 있는 [12]에서는 MDA와 Product Line을 기반으로 하는 방안을 제안하고 있으나 실제 생성방안에 대해서는 언급하고 있지 않다.

3. 서비스 온톨로지의 설계 및 생성 방안

이 장에서는 서비스 온톨로지를 설계하고 생성하기 위해 OWL-S와 UML간의 매핑에 대해 기술한다.

3.1 프로세스의 표현

서비스 온톨로지는 프로세스들의 행동양식을 기술함으로써 서비스가 어떻게 사용되는지를 기술한다. 즉, 서비스 온톨로지의 핵심은 프로세스이다. 프로세스는 IOPE(Inputs, Outputs, Preconditions, Effects)[3]로 기술되며 서비스를 나타내는 최소단위로 UML의 클래스로 표현할 수 있다. 따라서 프로세스가 가질 수 있는 속성들은 클래스와 연관된 속성들과 연결될 수 있다. 다음의 (그림 2)는 UML의 메타모델상의 클래스와 매핑된 프로세스의 메타모델을 나타낸다.

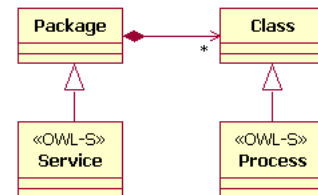


(그림 2) OWL-S의 프로세스와 UML의 매핑

이러한 메타모델을 바탕으로 프로세스를 클래스로 표현한 PIM을 설계할 수 있다. 이렇게 설계된 PIM은 OWL-S 뿐만 아니라 다른 구현언어, 즉 자바 또는 C++와 같은 언어에 최적화된 모델로도 변환될 수 있다. 즉, 실제 서비스를 수행하는 애플리케이션과 그 서비스를 기술하는 서비스 온톨로지를 하나의 모델로부터 이끌어 낼 수 있다. 이는 모델을 재사용하여 개발효율을 높이는 MDA의 특성을 웹 서비스의 개발에 적용할 수 있음을 나타낸다. 또한 서비스와 서비스 온톨로지를 하나의 모델에서 이끌어내기 때문에 서비스와 서비스 온톨로지 사이의 불일치 문제를 자연스럽게 해결할 수 있다.

3.2 OWL-S의 각 구성요소와 UML 메타 모델간의 매핑

서비스 프로파일은 프로세스들의 모델들을 묶어 하나의 추상적인 모델을 제공한다. 따라서 클래스로 표현된 프로세스들을 패키지로 묶어 사용할 수 있다. (그림 3)은 서비스 프로파일을 나타내기 위해 OWL-S의 구조물인 Service와 UML의 패키지와의 매핑을 나타낸다.



(그림 3) Service와 Package의 매핑

서비스 모델은 웹 서비스의 행동을 기술하는 온톨로지이다. 이러한 서비스 모델의 설계를 위해 우리는 UML의 활동도(Activity Diagram)를 사용한다. 활동도는 객체의 행동을 묘사하기 위한 UML의 표현 도구로써 서비스를 구성하는 프로세스들의 실행 순서와 데이터의 흐름 등을 효

과적으로 설계할 수 있다. 또한 여러 개의 복합 프로세스들로 구성된 복잡한 서비스를 쉽게 표현할 수 있으며, 나아가 설계 단계에서 다른 서비스와의 조합을 고려할 수 있다는 장점을 가진다[12].

서비스 모델은 프로세스와 프로세스들의 흐름을 제어하는 제어 구조물들로 구성된다. 이 구조물들은 각각 특성에 따라 <표 1>과 같이 UML 활동도의 구조물들과 매핑될 수 있다. 이 매핑관계는 OWL-S의 제어 구조와 UML 활동도 사이의 유사성[12]에서 기인한다.

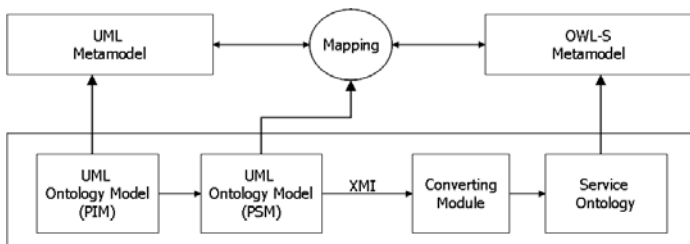
<표 1> 서비스 모델과 활동도 간의 매핑

Service Model	Activity Diagram
CopositeProcess	Swimlane
AtomicProcess	Action State
Sequence	Transition
Repeat	Self Transition
Choice	Decision
Split & Join	Synchronization

그라운딩은 웹 서비스의 실체화, 즉 실제 웹 서비스의 WSDL[7] 문서와의 매핑을 담당하는 온톨로지로서 프로세스 모델에서 표현된 웹 서비스와 실제 존재하는 WSDL로 표현된 웹 서비스간의 관계를 기술한다. 그라운딩 온톨로지는 WSDL 문서에 대한 매핑 정보를 나타내는 일련의 Grounding Class[3]들로 구성되어 있기 때문에 WSDL 문서로부터 쉽게 추출이 가능하다. 따라서 본 논문에서는 그라운딩 온톨로지를 WSDL 문서로부터 추출하는 방법을 채택했다.

3.3 변환 프로세스

설계된 모델을 정의된 매핑 관계를 기반으로 실제 서비스 온톨로지 변환하기 위해 우리는 (그림 4)와 같은 구조의 변환 프로세스를 정의하고 이를 수행하는 애플리케이션을 개발하였다.



(그림 4) 변환 프로세스의 블록 다이어그램

이 변환 프로세스는 다음과 같은 순서로 수행된다. 우선 UML 모델링 도구를 통해 웹 서비스의 PIM을 설계한다. 설계된 PIM은 MDA 도구에 의해 PSM으로 변환되며 변환된 PSM은 XMI 문서로 저장된다. XMI는 XML 문서이므로 XML Parser인 DOM[5]을 통해 실제 OWL-S 모델로 변환될 수 있다.

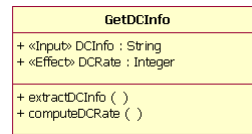
4. 실험 결과 및 검증

4.1 서비스 온톨로지의 설계 및 생성

우리는 실험을 위한 예로써 우리의 이전 연구 [10]에서

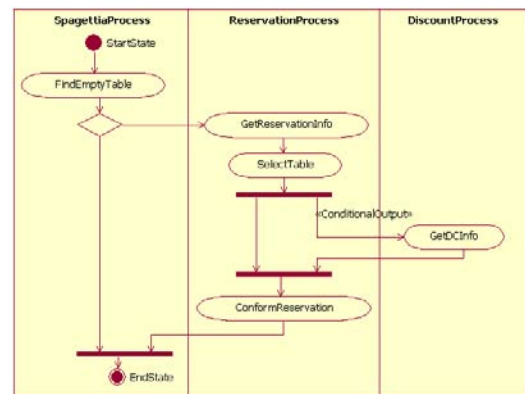
예로 들었던 레스토랑의 예약 서비스를 사용하였다.

우리는 레스토랑 예약 서비스를 구성하는 각 프로세스를 (그림 5)와 같은 PIM으로 설계하고 MDA 도구를 사용하여 PSM으로 변환하였다.



(그림 5) GetDCInfo의 PIM

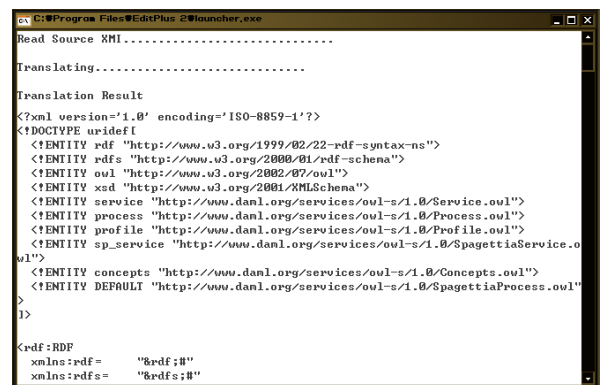
그리고 우리는 웹 서비스의 서비스 모델 온톨로지를 만들기 위해서 우리는 프로세스의 흐름을 정의하였다. (그림 6)은 프로세스의 흐름을 UML의 활동도로 모델링한 것이다.



(그림 6) Reservation Service Model

본 논문에서는 서비스 프로파일 온톨로지를 만들기 위한 방법으로 패키지를 이용한다. 이 패키지에서는 각 프로세스의 IOPE 정보와 프로세스 이름, 서비스 이름과 같은 정보를 이용하여 서비스 프로파일 온톨로지를 생성할 수 있다.

이렇게 정의된 서비스 온톨로지의 모델은 UML 도구에 의해 XMI 문서로 저장되며 저장된 XMI 문서는 변환 모듈을 통해 서비스 모델 온톨로지 변환된다. (그림 7)은 서비스 모델의 XMI 문서가 실제 서비스 온톨로지 변환되는 결과를 보여준다.



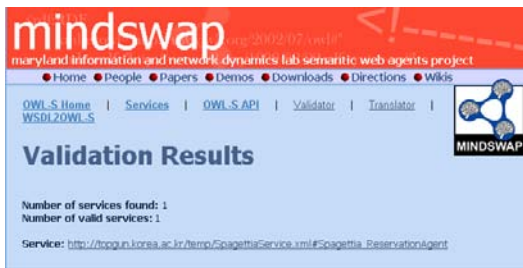
(그림 7) XMI 문서로부터 서비스 온톨로지로의 변환

본 논문에서 제안하는 방법에서는 서비스 온톨로지의 그라운딩을 추출하기 위해서 설계된 모델을 플랫폼에 맞는 모델로 변환하고 이를 실제 서비스로 구축함으로써 얻을

수 있는 WSDL 문서를 사용한다. 이를 위해 우리는 레스토랑의 예약 서비스를 .NET 플랫폼으로 구축하였으며 서비스 개발 도구로부터 WSDL 문서를 얻을 수 있었다. 우리는 미리 정의된 OWL-S Grounding Class와 WSDL 간의 매핑을 통해 서비스 개발 도구로부터 얻은 WSDL 문서에서 그라운드링 온톨로지를 추출하였다.

4.2 생성된 서비스 온톨로지의 검증

본 논문에서는 제안한 방법을 통해 생성된 서비스 온톨로지가 정확한지 여부를 검증하기 위해 [14]에서 제공하는 OWL-S Validator를 사용하였다. 이 도구는 [15]에서 제시한 온톨로지 검증 기준에 맞춰 OWL-S로 기술된 서비스 온톨로지를 파싱하여 문법적 오류가 없는지를 검증하고 서비스의 내용이 올바르게 기술되어 실제로 서비스가 가능한지를 검증한다. (그림 8)은 실험 결과로 생성된 서비스 온톨로지를 OWL-S Validator를 통해 검증한 결과를 보여준다.



(그림 8) 생성된 서비스 온톨로지의 검증 결과

5. 결론 및 향후 연구 과제

본 논문에서 우리는 MDA 방법론을 기반으로 지능형 웹 서비스를 위한 서비스 온톨로지의 설계 및 생성 방법을 제안하였다. 제안된 방법은 웹 서비스를 플랫폼에 독립적인 모델로 설계한 후 이 모델을 MDA 도구를 이용하여 플랫폼에 종속적인 모델로 변환한다. 그리고 이 모델들을 XMI를 통하여 실제 서비스 온톨로지로 변환한다. 우리는 제안된 방법을 검증하기 위해 [10]에서 예제로써 사용되었던 레스토랑 예약 서비스를 실제로 모델링하고 서비스 온톨로지를 생성하였다. 그리고 이 서비스 온톨로지를 검증 도구를 이용하여 검증함으로써 타당성을 입증하였다.

우리가 제안하는 방법은 서비스 온톨로지의 생성에 MDA를 사용함으로써 하나의 모델로부터 서비스 및 서비스 온톨로지를 동시에 생성할 수 있기 때문에 시간과 비용을 절약할 수 있다는 장점을 가진다. 또한 서비스와 서비스 온톨로지가 별개로 구축되어 완벽한 매핑을 보장할 수 없다는 문제를 해결할 수 있으며 배우기 어려운 서비스 온톨로지 마크업 언어 대신 소프트웨어 개발에 널리 사용되는 UML과 같은 방법론을 통해 기존의 개발자들도 쉽게 서비스 온톨로지를 생성할 수 있다. 그리고 플랫폼의 변화와 같은 외부적 변화에 보다 유연하게 대처할 수 있다.

향후 연구로서 다음의 두가지 과제가 있다. 첫 번째로는 그라운드링 온톨로지의 생성 방법에서 발생하는 조건부 입출력을 처리할 수 없는 문제를 해결하기 위한 방안을 모색할 계획이다. 현재 WSDL에서는 조건부 입출력을 표현

할 수 없다[11]. 향후 WSDL이 조건부 입출력의 표현을 지원하게 된다면 자연스럽게 해결될 수 있겠지만 현재로서는 생성된 그라운드링 온톨로지를 직접 수정하는 방법보다 효율적인 방법을 찾아야 할 것이다. 두 번째로는 규칙기반 웹 서비스 프레임워크[18]를 위해 서비스 온톨로지의 설계 단계에서 OCL(Object Constraint Language)[6]을 이용하여 온톨로지 모델에 규칙을 부여하는 방안을 연구할 계획이다. 이 프레임워크는 서비스 온톨로지에 규칙을 부여하여 자동적인 검색 및 조합을 가능케 하기 위한 목적을 가지고 있다. OCL로서 온톨로지 모델에 조건적인 제약사항을 기술할 수 있다면 규칙이 부여된 서비스 온톨로지를 만들 수 있는 가능성이 충분하다.

참조 문헌

- [1] MDA : <http://www.omg.org/mda/>
- [2] UML : <http://www.uml.org/>
- [3] OWL-S : <http://w3.org/2004/OWL-S/>
- [4] XMI : <http://www.omg.org/technology/documents/formal/xmi.htm>
- [5] DOM : <http://www.w3.org/DOM/>
- [6] OCL : <http://www.uml.org/>
- [7] WSDL : <http://www.w3.org/TR/2004/WD-wsdl20-primer-20041221/>
- [8] MOF : <http://www.omg.org/cgi-bin/doc?formal/00-04-03>
- [9] Stephen Cranfield, Stefan Haustein and Martin Purvis, UML-Based Ontology Modeling for Software Agents, Proc. of Ontologies in Agent Systems Workshop, pp. 21-28, 2001
- [10] 양진혁, 민재홍, 이윤수, 김태석, 정인정, 지능형 e-비즈니스를 위한 플랫폼에 관한 연구: 시맨틱 웹 서비스 아키텍처, 21회 한국정보처리학회 춘계학술발표대회 논문집 11권 1호, pp.369-372 2004. 5
- [11] 지능형 웹 서비스 표준 기술 동향 및 국내 도입 방안 연구 보고서, 한국 전산원, 2004. 4
- [12] Gerald C. Gannod, John T. E. Timm, An MDA-based Approach for Facilitating Adoption of Semantic Web Service Technology, Proceedings of the 8th IEEE Enterprise Distributed Object Computing Conference Workshop on Model-Driven Semantic Web, September 2004
- [13] Zakaria Maamar, Boualem Benatallah, Wathiq mansoor, Service Chart Diagrams - Description & Application, The Twelfth International World Wide Web Conference, pp.43-49, 2003
- [14] OWL-S Validator : <http://www.mindswap.org/2004/owl-s/validator/>
- [15] Bijan Parsia, Evren Sirin, and Aditya Kalyanpur. Debugging OWL ontologies. In The 14th International World Wide Web Conference, Chiba, Japan, May 2005