

웹 서비스 발견을 위해 프로세스 정보를 기술하는 온톨로지

Ontology describing Process Information for Web Services Discovery

유정연(Jeong-Youn Yu)*, 이규철(Kyu-Chul Lee)**

초 록

지금까지 시맨틱 웹 서비스 발견은 WSMO나 OWL-S의 프로파일을 이용하여 수행되어 왔다. 그러나 이 연구들은 서비스의 이름과 input/output만을 이용하여 검색함으로써, '배송 후 지불을 하는 도서 판매 서비스를 찾아라.'와 같이 서비스의 내부 정보에 대한 질의를 수행할 수 없다. 본 논문에서는 이러한 문제를 해결하기 위해 TM-S 온톨로지와 TMS-QL을 제시한다. TM-S 온톨로지는 프로세스 정보를 이용하여 서비스의 행동을 기술하며, 3가지 하위 온톨로지인 프로세스 시그니처 온톨로지, 프로세스 구조 온톨로지, 프로세스 개념 온톨로지 구성되어 있다. TMS-QL은 사용자가 TM-S 온톨로지에 대해 질의 요구를 기술할 수 있는 언어를 제공한다.

ABSTRACT

Until now, most semantic web service discovery research has been carried out using either Web Service Modeling Ontology (WSMO) or a profile of OWL-based Web Service ontology (OWL-S). However, such efforts have focused primarily on service name and input/output ontology. Thus, the internal information of a service has not been utilized, and queries regarding internal information such as 'Find book-selling services allowing payment after delivery' are not addressed. This study outlines the development of TM-S (Topic Maps for Service) ontology and TMS-QL (TM-S Query Language), two novel technologies that address the aforementioned issues in semantic web service discovery research. TM-S ontology describes the behavior of services using process information and consists of three sub-ontologies: process signature ontology, process structure ontology and process concept ontology. TMS-QL allows users to describe service discovery requests.

키워드 : 시맨틱 웹 서비스, 서비스 발견, 온톨로지, 토픽 맵

Semantic Web Service, Service Discovery, Ontology, Topic Maps

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원 사업(IITA-2006-C1090-0603-0031)의 연구결과와 한국산업기술평가기원이 지정한 지역협력연구센터(RRC)인 충남대학교 소프트웨어 연구 센터의 지원으로 수행된 과제의 결과입니다.

* 충남대학교 소프트웨어연구센터 전임연구원

** 충남대학교 전기정보통신공학부 교수, 교신저자

1. 서 론

웹 서비스의 기본 표준들에서 서비스의 기술(description)은 WSDL(Web Service Description Language), 발견(discovery)은 UDDI(Universal Description Discovery and Integration)을 이용한다[6]. 서비스 제공자는 WSDL을 이용하여 서비스를 기술하고, UDDI 레지스트리(registry)에 서비스를 등록한다. 서비스 요구자는 UDDI 레지스트리를 통해 원하는 서비스를 검색한다. WSDL은 서비스의 구문 정보(syntactic information)만을 기술하고, UDDI는 키워드를 기반으로 서비스를 검색한다. 그러나 WSDL과 UDDI를 이용한 서비스 검색은 만족하지 않는 검색 결과를 제공한다. 예를 들어, 사용자가 "가격을 입력 받아 구매가 가능한 자동차들을 출력하는 자동차 판매 서비스를 찾아라."라고 요청했다고 가정하자. 이 질의는 가격을 입력(input)받고, 자동차를 출력(output)하는 자동차 판매 서비스를 검색하는 것이다. 그러나 현재 웹 서비스는 서비스의 이름과 설명 정보만을 이용하여 검색하므로 관련 있는 서비스들을 검색하지 못한다.

시맨틱 웹 서비스(Semantic Web services)는 위와 같은 현재 웹 서비스의 발견을 보완하기 위한 하나의 수단으로 제안되었다. 즉, 시맨틱 웹 서비스는 시맨틱 웹을 이용하여 웹 서비스를 기술한 것으로, 지능적 웹 서비스 발견을 제공하며, 자동화된 조합 및 상호 운용이 가능하도록 한다[4, 14]. 시맨틱 웹을 가능하게 하는 기술은 온톨로지(Ontology)이다. 이는 개념과 그들 간의 관계들로 정의하며, 특정 도메인에 관련된 단어

들을 계층적으로 표현하고 이를 확장하여 추론 규칙을 가짐으로써 온톨로지를 표현한다[8, 9, 15]. 온톨로지는 기존의 웹 서비스가 가지는 발견의 문제점을 해결할 수 있다. 앞에서 언급한 질의에 대해 input/output에 대한 온톨로지를 정의하고 검색하면, "SUV", "Sedan"과 같이 자동차에 관련된 것을 출력(output)하는 자동차 판매 서비스들을 모두 검색할 수 있다. 시맨틱 웹 서비스 발견은 WSMO와 OWL-S의 서비스 프로파일(Service Profile)을 이용하여 검색되어 왔다. 그러나 이 발견은 서비스 이름과 input/output에 관련된 온톨로지만을 이용하여 검색한다. 이것은 "배송 후 결제를 하는 도서 판매서비스를 찾아라."와 같은 질의가 요청될 때, 서비스의 내부 정보를 이용하지 않아 사용자가 원하는 서비스들을 검색하지 못하거나 관련 없는 서비스들이 검색된다.

본 논문은 서비스의 내부 정보를 이용하여 검색하기 위해 필요한 TM-S 온톨로지와 TMS-QL을 개발하였다. 서비스는 웹에서 접근이 가능한 프로그램으로 하나의 프로세스 또는 여러 개의 하위 프로세스들로 표현될 수 있다. 사용자는 최상위 프로세스를 이용하여 서비스의 정보를 검색할 수 있다. TM-S 온톨로지는 프로세스 정보를 이용하여 서비스의 내부 정보를 상세히 기술한 것이며, TMS-QL은 사용자가 TM-S 온톨로지를 이용하여 서비스를 검색하는 질의 언어이다.

본 논문은 다음과 같이 구성되어 있다. 제2장은 시맨틱 웹 서비스의 표준들과 발견 방법들에 대해 살펴보고, 현재의 시맨틱 웹 서비스 발견이 가지는 문제점들을 분석한다.

제 3장은 제 2장에서 언급한 문제점들을 해결하기 위해 프로세스 정보를 온톨로지 모델링한다. 이를 위해 웹 서비스 발견에 필요한 프로세스 정보들을 추출하고, 이 정보들을 온톨로지 표현하기 위한 방법들을 알아본다. 제 4장은 시맨틱 웹 언어인 Topic Maps를 이용하여 프로세스 정보를 표현한 TM-S 온톨로지를 제안하고, 제 5장은 TMS-QL을 이용하여 서비스를 검색하는 예를 보여준다. 제 6장은 연구 내용을 요약하고, 앞으로의 연구 방향에 대해 언급한다.

2. 관련 연구

2.1 시맨틱 웹 서비스 표준 및 발견 연구들

시맨틱 웹 서비스를 기술하는 대표적인 표준으로는 WSMO, OWL-S, SWSO(Semantic Web Services Ontology)가 있다. WSMO는 e-워크(e-Work)와 e-전자상거래(e-commerce)에서 서비스들 간의 통합 문제를 해결하는 것을 목표로, 온톨로지, 목적(goal), 웹 서비스 미디어이터(mediator)와 같은 구성 요소를 이용하여 온톨로지를 기술한다[20]. 온톨로지는 다른 요소들을 기술하기 위해 필요한 용어와 정형화된 의미론(formal semantics)을 제공한다. 목적은 사용자가 찾고자 하는 서비스의 정보들을 정의한다. 웹 서비스는 서비스에 대한 시맨틱 기술이다. 미디어이터는 구성 요소들 간의 이질적 문제를 해결하는 연결자 역할을 한다.

OWL-S는 웹 서비스의 속성(properties)과

능력(capabilities)을 기술하는 온톨로지이다 [11]. 이 표준은 Stanford, SRI, CMU 등 미주를 중심으로 공동으로 개발되고 있으며, 2006년 2월에 버전 1.2를 공표하였다. OWL-S는 3가지 온톨로지인 서비스 프로파일, 프로세스 모델(Process Model), 서비스 그라운드(Service Grounding)으로 구성되어 있다. 이 온톨로지들은 웹 서비스의 생명 주기에서 서로 다른 활동(activities)에 적용된다[13]. 서비스 프로파일은 서비스 제공자가 서비스를 공표하고, 서비스 요구자가 서비스를 검색하는데 사용된다. 만일, 찾은 서비스가 사용자의 요구 사항에 맞는지를 알려면, 프로세스 모델을 이용하여 검토한 후 서비스를 선택할 수 있다. 이 프로세스 모델은 서비스의 시뮬레이션, 검증, 모니터링, 다중 서비스들과의 조합에 이용된다. 서비스 그라운드는 서비스 접근에 관련된 서비스 호출 및 상호 운영에 이용된다.

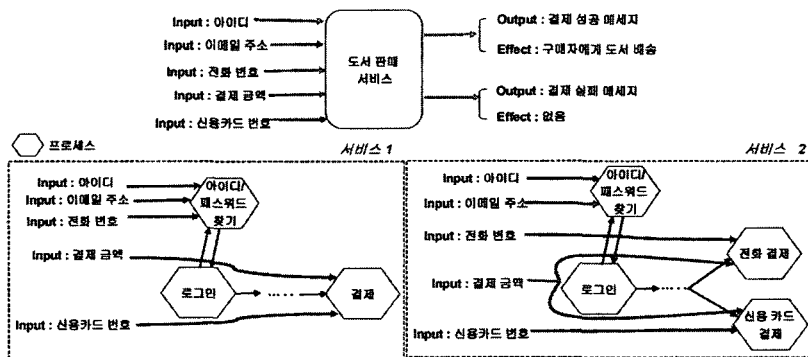
SWSO는 시맨틱 웹 서비스를 위한 개념적 모델을 정의한다[21]. FLOWS(First-order Logic Ontology for Web Services)은 SWSL-FOL(First Order Logic SWSL)로 표현된 SWSO로써, 시맨틱 웹 서비스를 선언하고 여러 환경에서 시맨틱 웹 서비스의 추론을 가능하게 한다. FLOWS에서 웹 서비스의 기술은 PSL(Process Specification Language)의 메커니즘을 기반으로 한다. FLOWS의 구조는 OWL-S에서 제안한 것과 매우 유사하다. 이것은 서비스 기술자(Service Descriptor), 프로세스 모델, 그라운딩으로 구성되어 있다. 이들 중에서 서비스 기술자는 이름, 저자, 연락 정보, URL 등과 같이 서비스를 발견하고 매칭하는데 이용되는 웹 서비스에 관

한 기본 정보를 제공한다.

시맨틱 웹 서비스에서 발견 연구로는 WSMO 명세서, Matchmaker 프로젝트, METEOR-S 프로젝트에서 진행되어 왔다. WSMO 명세서에서 서비스의 발견은 서비스 요구자가 기술한 목적을 만족하는 웹 서비스를 찾는다[10]. 목적과 웹 서비스간의 기본적인 매칭 모델은 output과 effect를 고려하여 서비스를 검색한다. Matchmaker 프로젝트와 METEOR-S 프로젝트는 OWL-S의 프로파일 온톨로지를 이용한다. Matchmaker 프로젝트는 OWL-S 매칭 엔진을 이용하여 서비스 프로파일로 정의된 서비스들을 발견한다[17]. OWL-S 매칭 엔진은 서비스의 input과 output의 매칭 정도를 비교하여 원하는 서비스를 검색한다. METEOR-S 프로젝트는 e-워크플로우에서 서비스를 추가 시, 서비스를 발견하는 방법에 대해 연구한 것이다[7]. 이 연구는 서비스 이름, 설명, input/output, precondition/effect, 품질(quality)을 이용하여 서비스를 검색한다.

2.2 기존 연구의 문제점들

기존의 연구들은 서비스를 발견하는데 있어 몇 가지 문제점을 가지고 있다. 첫째, 사용자는 서비스 이름, input/output과 같은 서비스의 외부 정보만을 이용하여 검색한다. 예를 들어, 사용자가 "전화번호를 입력 받아 결제 할 수 있는 도서 판매 서비스를 찾아라."라고 요청했다고 하자. 기존 연구에서는 <그림 1>의 상단과 같이 전화번호를 input으로 받고, 결제메시지를 output하는 도서 판매 서비스들이 검색 될 것이다. 그러나 모든 결과가 사용자가 원하는 서비스는 아니다. 같은 input들과 output을 가진 도서 판매 서비스들이라도, 서비스 1은 전화번호가 아이디/패스워드 찾기에 사용되며, 서비스 2는 전화번호가 전화 결제에 사용 된다. 이때, 서비스 2만 사용자의 요구 사항을 만족하는 결과가 된다. 그러므로 사용자가 원하는 서비스만을 검색하기 위해서는 input/output과 이들과 관련된 프로세스를 함께 이용하여



<그림 1> 서비스의 외부 정보를 이용한 검색의 문제점

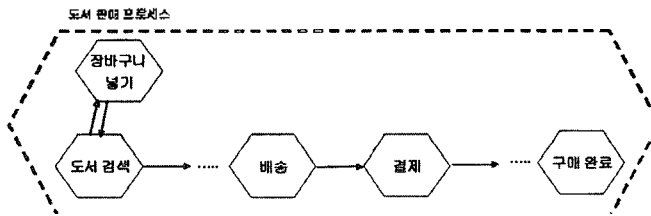
검색해야 한다.

둘째, 서비스의 수행 과정에 대한 상세 정보를 알 수 없다. 예를 들어, "배송 후 결제가 가능한 도서 판매 서비스를 찾아라."라고 한다고 하자. 이와 같은 질의는 <그림 2>와 같이 서비스를 구성하는 프로세스들의 내부적인 동작들을 보아야만 알 수 있다. 그러나 현재 연구들은 서비스 제공자가 정의한 서비스의 기술 정보에 의존하여 검색한다.

세 번째, 프로세스의 단순한 개념만을 검색한다. 사용자가 "전자 상점을 통해 고객에게 책을 판매하는 프로세스를 찾아라."라고

여 검색한다.

첫 번째 문제는 프로세스 이름과 그와 연계되어 있는 input/output 이름을 정의한 온톨로지를 이용하면 해결할 수 있다. 두 번째 문제는 서비스의 내부 동작인 프로세스들의 제어 구조를 기술한 온톨로지를 이용하면 해결 가능하다. 시맨틱 웹 서비스 표준 중에서 OWL-S의 프로파일과 SWSO의 FLOWS 서비스 기술자는 프로세스와 서비스의 내부 정보를 정의하고 있지 않다. WSMO는 서비스의 능력과 오케스트레이션 인터페이스 (Orchestration Interface)를 이용하

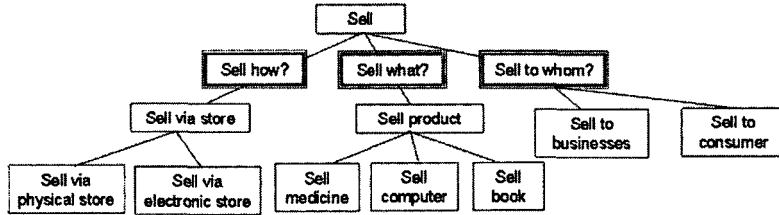


<그림 2> '배송 후 결제'를 수행하는 도서 판매 프로세스 예

하였다고 하자. 이 질의는 판매 프로세스에 대해 판매 방법(sell how), 판매 대상(sell to whom), 판매 종류(sell what) 관점에서 정의한 개념들을 이용하여 검색한다. 즉, 판매에 대해 판매 방법관점인 전자 상점, 판매 대상(sell to whom) 관점인 고객, 판매 종류(sell what) 관점으로는 책의 개념을 가지는 프로세스를 찾는다. 이것은 <그림 3>과 같이 프로세스의 개념들에 대해 다양한 관점들을 이용하여 정의한 온톨로지를 이용하여 해결할 수 있다. 그러나 기존의 연구들은 제품에 대해 단순하게 분류한 온톨로지만을 이용하

여 서비스의 내부 동작을 정의하지만, 아직 명세서에 대한 기술이 완료되지 않아 사용할 수 없다. 그러나 OWL-S의 프로세스 모델과 SWSO의 FLOWS 프로세스 모델은 서비스의 시뮬레이션, 검증, 조합을 목적으로 사용되지만, 프로세스와 제어 구조를 이용하여 서비스의 내부 동작을 정의하고 있다. 이와 같이 구성된 온톨로지를 서비스 발견에 이용하면 첫 번째 문제와 두 번째 문제를 해결할 수 있다.

세 번째 문제는 프로세스 개념을 다양한 관점으로 정의한 온톨로지를 이용하면 해결



〈그림 3〉 판매 개념에 대한 온톨로지 예

이 가능하다. OWL-S의 ProfileHierarchy나 외부 분류 정보인 UNSPSC과 NAICS은 제품에 대한 단순한 분류 온톨로지만을 제공한다. 그리고 WSMO와 SWSO는 프로세스의 개념들에 대한 온톨로지가 정의되어 있지 않아 이 문제를 해결할 수 없다. 그러나 MIT (Massachusetts Institute of Technology)의 Process Handbook 프로젝트는 서비스 모델을 검색하기 위해 인덱스를 제공한다[5]. 이 인덱스는 서비스 모델을 계층적으로 분류하고 관련된 분류 항목들에 대해 번들(bundle)을 이용하여 결합하는 구조를 제공한다. 이 인덱스 구조와 같은 형태를 가지는 프로세스 개념을 정의한 온톨로지는 이 문제를 해결하는데 바람직하다. 본 논문에서는 위의 세 가지 문제들을 해결하기 위해 프로세스 정보를 표현한 온톨로지를 제안한다.

3. 프로세스 정보를 표현하는 온톨로지 모델링

3.1 프로세스 정보를 표현하기 위한 온톨로지 요소 분석

프로세스 정보는 3가지의 세부 온톨로지

로 표현되며, 온톨로지 모델링에 필요한 요소들은 다음과 같다.

- 프로세스 시그니처 온톨로지: 이 온톨로지는 프로세스와 input/output에 대한 텍스트 정보를 기술한다. 즉, 프로세스 이름과 그와 연계되어 있는 input/output 이름에 대한 정의를 요구한다.
- 프로세스 구조 온톨로지: 프로세스 구조 온톨로지는 서비스의 내부 동작을 기술하는 것이다. 이것은 순차, 분기, 분기+결합 등과 같이 프로세스 구조를 표현하기 위해 필요한 제어 구조의 정의가 요구된다.
- 프로세스 개념 온톨로지: 이 온톨로지는 프로세스 개념을 다양한 관점에서 정의한 것이다. 이것은 프로세스 개념들에 대한 정의, 개념들 간의 계층적인 분류, 분류에 대한 다양한 관점을 정의할 수 있는 요소가 필요하다.

이 온톨로지 요소들 중에 일부는 이미 기존에 정의되어 있거나, 유사한 형태의 구조로 존재한다. 〈그림 4〉는 프로세스 정보를 표현하기 위해 필요한 온톨로지 요소들을 기존에 정의된 모델과 매핑한 것이다.

프로세스 시그니처 온톨로지와 프로세스

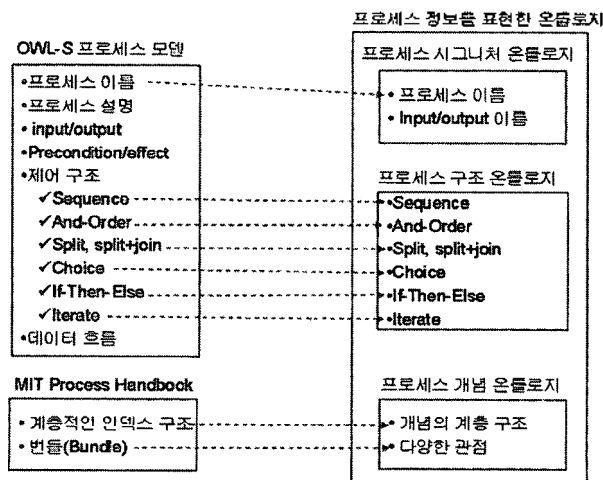
구조 온톨로지에서 대부분의 구성 요소들은 기존의 OWL-S 프로세스 모델과 SWSO의 FLOWS 프로세스 모델에 정의되어 있다. SWSO의 FLOWS는 OWL-S를 기반으로 확장한 것이다. 현재, OWL-S는 다른 표준들보다 완성된 온톨로지를 제공하나, SWSO의 FLOWS는 아직 개발 초기 단계에 있다. 그러므로 본 논문에서는 OWL-S에서 제공하는 온톨로지를 기반으로 모델링을 하였다.

프로세스 시그니처 온톨로지의 프로세스 이름은 OWL-S 프로세스 모델에 정의된 프로세스 이름과 매핑이 된다. 프로세스 구조 온톨로지의 제어 구조정보는 OWL-S 프로세스 모델의 7가지 제어 구조 형태와 매핑된다. 그러나 OWL-S 프로세스 모델에 정의되어 있지 않은 input/output 이름은 새로이 추가해야 한다. 프로세스 개념 온톨로지에 대한 정의는 Process Handbook 프로젝트의 형태 구조를 이용하여 모델링 할 수 있다. 즉, 개념간의 계층적인 분류는 Process

Handbook의 인덱스 구조와 같은 형태로 모델링하며, 분류에 대한 관점은 관련된 항목들을 그룹화 하는 번들과 같은 형태로 모델링 할 수 있다. 본 논문에서는 프로세스 정보를 표현하는 온톨로지를 위해 이미 구축된 온톨로지들은 활용하고 필요한 요소들은 새로이 추가함으로써 효율적인 모델링을 구축한다.

3.2 OWL을 이용하는 방법

프로세스 시그니처 온톨로지와 프로세스 구조 온톨로지의 대부분은 OWL로 표현되어 있다. 그러므로 OWL을 이용하여 프로세스 개념 온톨로지를 정의하는 방법을 고려해 볼 수 있다. OWL은 RDF(Resource Description Framework)와 RDFS(RDF Schema) 모델을 확장한 것으로, 온톨로지의 구축에 필요한 클래스와 속성, 이들 간의 관계를 정의하는 어휘들을 제공하고 있다[12]. 그러나



〈그림 4〉 기존 모델과 본 논문에서 제안하는 온톨로지와의 매핑 관계

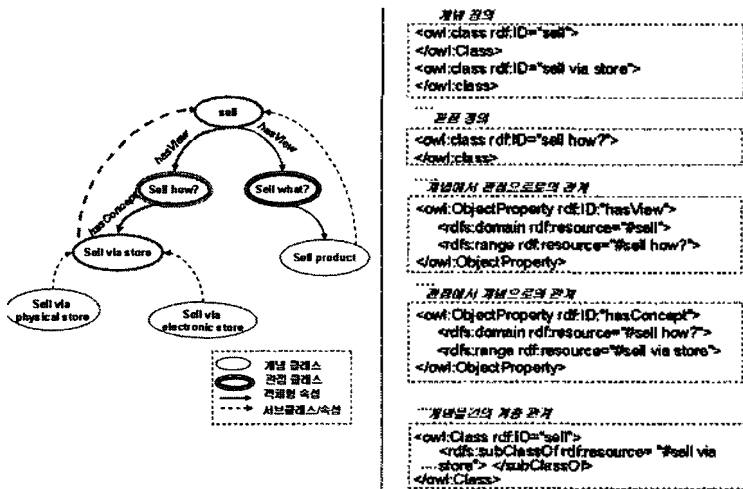
OWL은 프로세스 개념 온톨로지에 정의된 관점을 표현할 수 있는 개념과 문법을 가지고 있지 않다. 그러므로 프로세스 개념 온톨로지를 표현하기 위한 클래스(class)들과 속성(property)들을 추가로 정의해야 한다.

〈그림 5〉는 판매에 대한 개념에 대해 OWL 기반의 모델로 정의한 것이며, 〈그림 4〉에 대해 OWL 문법을 이용하여 표현한 예이다. 우선 개념 클래스들과 관점 클래스들을 각각 정의한다. 즉, 'sell', 'sell via store', 'sell via electronic store' 등과 같은 개념 클래스와 'sell how?', 'sell what?' 과 같은 관점 클래스를 정의된다. 'hasView' 객체형 속성을 이용하여 개념에서 관점으로서의 관계를 표현한다. 'hasConcept' 객체형 속성은 관점에서 개념으로서의 관계를 표현한다. 상위 개념과 하위 개념간의 계층 구조를 표현하기 위해 'rdfs:subClassOf' 를 이용한다. 그러나 OWL은 2진 관계 표현법을 제공하므로 개념들간

또는 개념과 관점 간에 여러 개의 관계 정의가 필요하다. 이것은 개념들에 대한 관계 정의가 분산되어, 프로세스 개념들의 계층 구조를 이해하는데 어렵게 한다. 또한, 같은 개념에 대한 관점의 수가 많아질수록, 개념에 대한 발견이 복잡해진다. 이것은 만일 개념과 개념 간에 n개의 관점이 존재할 경우, 최소 2ⁿ ~ 최대 2ⁿ⁺¹개의 모델 표현이 필요하기 때문이다.

3.3 다른 시맨틱 언어를 이용하는 방법

온톨로지를 표현하는 다른 대안으로서, 새로운 시맨틱 웹 언어를 이용하여 프로세스 시그니처 온톨로지, 프로세스 구조 온톨로지, 프로세스 개념 온톨로지를 표현하는 방법을 생각해 볼 수 있다. 온톨로지 언어 중에 Topic Maps는 지식 구조를 기술하고 정



〈그림 5〉 판매 개념에 대한 OWL 기반의 모델

보 자원과의 연관 관계를 기술하여 온톨로지를 정의하는 개념적 모델을 제공한다[18]. 이 모델은 2000년 국제표준화기구인 ISO(the International Organization for Standardization) 표준으로 승인되었다. 최근에는 웹상에서 Topic Maps의 교환을 목적으로, XML(eXtensible Markup Language)을 기반으로 Topic Maps의 모델을 정의한 XTM(XML Topic Maps) 명세서를 발표하였다[19]. Topic Maps는 3가지의 기본 요소로 구성되어 있다. Topic, Association, Occurrence이다. Topic은 사람, 사물, 개념 등과 같이 어떤 대상을 표현한다. Topic은 한 개 이상의 Topic 타입과 이름을 가진다. Association은 2개 이상의 Topic들 간에 관계를 정의한다. Association에 참여하는 Topic들은 역할을 가진다. Occurrence는 Topic에 관련된 정보 자원을 정의한다.

이러한 Topic Maps의 개념 모델은 프로세스 시그니처 온톨로지와 프로세스 구조 온톨로지의 지식 구조를 잘 표현할 수 있다. Topic은 프로세스와 input/output, 프로세스의 제어 구조를 표현하며, Association은 프로세스와 input/output, 프로세스의 제어 구조와 구성 요소들 간의 관계를 표현한다. 또한, 프로세스 개념 온톨로지에 대해 OWL를 이용한 방법보다 체계적인 온톨로지 표현이 가능하다. Topic은 개념을 표현하며, Association은 개념들간의 계층 관계를 표현한다. Association이 가지는 다중 관계와 Scope의 특성은 여러 개의 개념들과의 관계를 정의하고 이 개념들간의 다양한 관점을 표현하는데 효율적인 표현 능력을 제공한다. 그러므로 본 논문에서는 Topic Maps를 이용하여

프로세스 정보를 표현하는 TM-S 온톨로지를 개발하였다.

4. TM-S 온톨로지

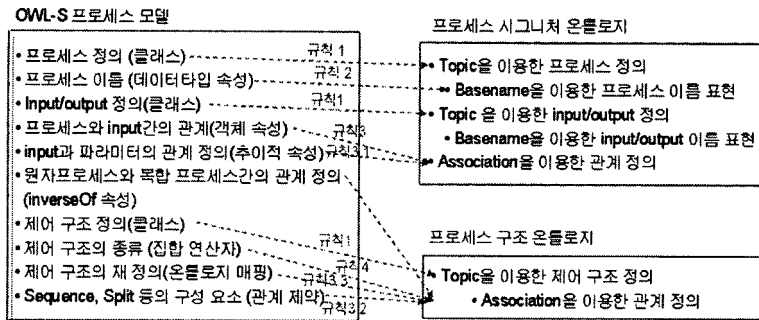
TM-S 온톨로지는 3가지로 나누어 표현한다. 첫째, 이미 구축된 온톨로지를 Topic Maps 기반의 프로세스 시그니처 온톨로지와 프로세스 구조 온톨로지로 활용하기 위해, OWL-S를 Topic Maps를 이용하여 변환한다. 둘째, 프로세스 개념 온톨로지를 시맨틱 웹 기반의 온톨로지로 활용하기 위해 Topic Maps를 이용하여 표현한다. 마지막으로, 프로세스와 프로세스 개념 온톨로지와의 관계 표현하여 프로세스에 대한 다양한 의미를 나타낸다.

4.1 프로세스 시그니처 온톨로지와 프로세스 구조 온톨로지의 표현

OWL-S의 온톨로지를 TM-S 온톨로지로 변환하기 위해서는 OWL-S에 정의된 정보들을 Topic Maps의 특성에 맞게 변환해야 한다. 그리고 만일 E가 OWL-S의 온톨로지라면, Topic Maps를 이용하여 E와 동등한 온톨로지로 정의해야 한다. 이 변환 작업은 <그림 6>과 같은 변환 규칙들을 이용하여 표현된다.

● 변환 규칙 1

프로세스 시그니처 온톨로지와 프로세스 구조 온톨로지를 기술하기 위해 필요한 모든 개념들은 Topic으로 정의한다. 예를 들어,



〈그림 6〉 적용된 변환 규칙들

프로세스, input/output, 프로세스의 제어 구조 등은 모두 Topic으로 정의한다.

● 변환 규칙 2

특정 개념의 이름은 그 개념을 정의한 Topic의 Basename으로 정의한다. 예를 들어, 프로세스의 이름은 Topic '프로세스'의 Basename으로 정의한다.

〈그림 7〉은 Topic 'sellBookProcess' 프로세스를 정의하고, 프로세스의 이름을 기술한 예이다. Topic 'sellBookProcess'는 <baseName> 태그 안에 '도서 판매 서비스'라는 프로세스 이름을 정의한다.

〈그림 7〉 XTM을 이용하여 'sellBookProcess' 프로세스를 정의한 예

● 변환 규칙 3

OWL에서 클래스들 간의 관계는 속성으로 표현한다. 이 관계는 Topic Maps에서 Association을 이용하여 동일하게 표현할 수 있다. 만일 클래스 X, Y에 대한 관계 P를 정의한다면, 다음과 같은 변환 규칙을 적용한다.

OWL : P (X, Y) ≡ P (Rx: X, Ry: Y): Topic Maps

OWL-S에서 두 개념 간의 관계는 P(X,

```

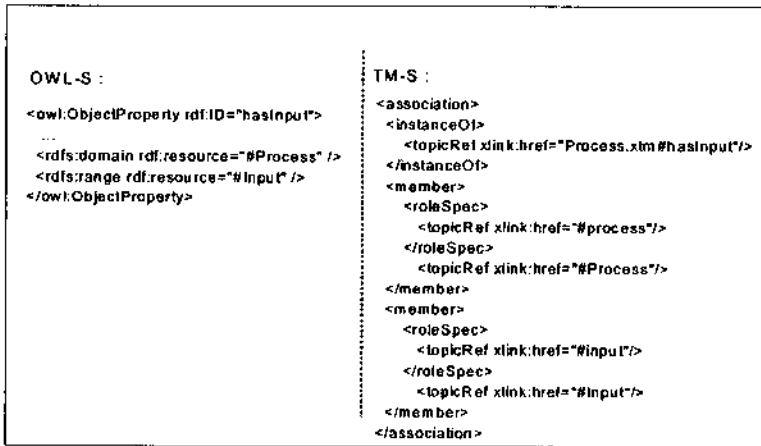
<topic id="SellBookProcess"> Topic 정의
  <instanceOf>
    <topicRef xlink:href="Process.xml#CompositeProcess" />
  </instanceOf>
  ...
  <baseName>
    <baseNameString>도서 판매 프로세스</baseNameString> 프로세스의 이름
  </baseName>
  <occurrence>
    <instanceOf>
      <topicRef xlink:href="#Description" />
    </instanceOf>
    <resourceData> This process is the case of selling book. </resourceData>
  </occurrence>
</topic>
    
```

〈그림 7〉 XTM을 이용하여 'sellBookProcess' 프로세스를 정의한 예

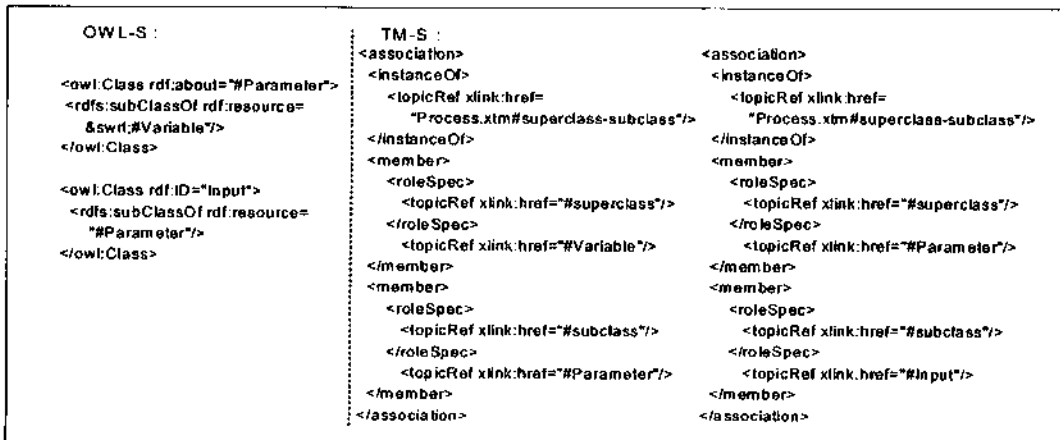
Y)로 표현할 수 있다. X는 영역 클래스를 나타내는 변수이다. Y는 범위 클래스를 나타내는 변수이다. P는 관계를 정의하는 속성이다. P(X, Y)는 X에서 Y로의 관계를 정의하며, X가 Y와 P 관계를 가진다는 것을 의미한다. Topic Maps는 (Rx: X, Ry: Y)로 표현할 수 있다. P는 Association 타입으로 표현한다. X, Y는 Association에 참여하는 Topic들이다. Rx는 Association에서 Topic X의 역할

을 나타낸다. Ry는 Association에서 Topic Y의 역할을 나타낸다. 예를 들어, 'Process'와 파라미터 변수 'Input' 간의 관계를 정의해 보자. 'Process'는 파라미터 변수 'Input'을 가질 수 있다. 이것은 아래와 같은 규칙을 적용하여 'Process'와 'Input' 간의 관계를 정의할 수 있다. <그림 8>은 OWL-S와 TM-S를 이용하여 이 관계를 표현한 것이다.

OWL-S: hasInput(Process, Input)



<그림 8> 프로세스와 input간의 관계 정의



<그림 9> 변수와 input 간의 추론 규칙

≡ hasInput (process: Process, input: Input)
: TM-S

● 변환 규칙 31 추론 관계

OWL에서 속성의 특성은 속성에 대한 항상된 추론을 제공한다. OWL-S에서는 추이적 속성을 이용하여 계층 관계를 추론한다. X, Y, Z 에 대한 추이적 속성은 다음과 같다.

OWL: $P(X, Y) \wedge P(Y, Z) \Rightarrow P(X, Z)$
 $\equiv P(Rx: X, Ry: Y) \wedge P(Ry: Y, Rz: Z)$
 $\Rightarrow P(Rx: X, Rz: Z): \text{Topic Maps}$

X가 Y와 P관계를 가지면서 Y가 Z와 P의 관계를 가지면, X와 Z는 P의 관계를 가지는 것으로 추론할 수 있다. OWL-S에서는 SubClassOf속성과 SubPropertyOf속성을 이용하여 추이적 속성을 나타낸다. SubClassOf는 클래스간의 상하위 관계를 추론하며, SubPropertyOf는 속성들 간의 상하위 관계를 추론한다. Topic Maps는 클래스들 간의 상하위 관계를 추론하기 위해 일반적으로 Superclass-Subclass 타입의 Association을 사용하고 있다. 또한, 속성들 간의 상하위 관계는 SuperProperty-SubProperty 타입의 Association을 이용한다. <그림 9>는 변수(variable)와 input 간의 추론 관계를 정의한 것이다. 파라미터는 변수의 하위 클래스이며, input은 파라미터의 하위 클래스이다. 그러므로 input은 변수의 하위 클래스에 속함을 추론할 수 있다.

그 외에 추론 관계를 정의하는 속성으로 기능적 속성(FunctionalProperty), 역 기능적 속성(InverseFunctionalProperty), 대칭적 속성

(SymmetricProperty)이 있다. 이 속성들은 OWL-S에는 정의되어 있지 않지만, Topic Maps를 이용하여 변환할 수 있다. 기능적 속성은 단일한 값을 가지는 속성을 정의하는 것으로, Topic Maps에서는 스키마 언어로 표현을 할 수 있다. 이 스키마 언어는 Topic Maps로 정의된 정보들을 제약하는데 사용된다. 이와 마찬가지로, 기능적 속성의 역인 역기능적 속성도 Topic Maps의 스키마 언어로 표현이 가능하다. 대칭적 속성은 클래스 X와 Y에 대한 대칭적인 관계를 정의한 것이다. 즉, P(X, Y)에서 클래스 X에서 Y로 P 관계를 가지면, 클래스 Y에서 X로도 P 관계를 가짐을 나타낸다. 이것은 규칙 3을 그대로 적용하여 다음과 같이 Topic Maps로 변환이 된다.

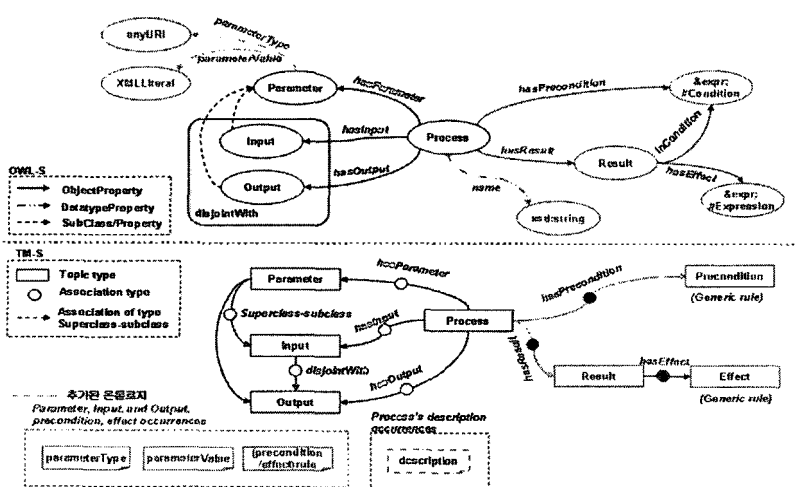
OWL: $P(X, Y) \Rightarrow P(Y, X) \equiv P(Rx: X, Ry: Y)$
 $\Rightarrow P(Ry: Y, Rz: Z): \text{Topic Maps}$

<그림 10>은 규칙 1에서 규칙 31까지를 적용하여 OWL-S 프로세스 모델을 프로세스 시그니처 온톨로지(굵은 선)로 표현하기 위해 변환한 것이다.

● 변환 규칙 3.2 관계 제약

OWL은 특정 관계에 적용되는 클래스의 범위를 제약할 수 있다. P(X, Y)에서 클래스 Y값의 제약은 allValuesFrom, someValuesFrom, hasValue를 사용하여 표현한다. 다음은 allValuesFrom을 사용하여 Y 값을 제한한 것이다.

OWL: $P(X, Y) \wedge \{Y = \text{allValuesFrom}(C)\}$
 $\equiv P(Rx: X, \text{allValuesFrom}: C) : \text{Topic Maps}$



<그림 10> OWL-S 프로세스 모델과 TM-S의 프로세스 시그니처 모델

X는 Y와 P의 관계를 가지며, Y는 C에 정의된 모든 값을 가지는 것을 의미한다. Topic Maps에서 특정 개념에 대한 제약은 Association에 참여하는 Topic Y의 역할을 이용하여 표현한다. 즉, Topic에는 Y의 값인 C를 정의하고, Topic의 역할에는 Y의 제약 사항인 allValuesFrom을 정의한다. <그림 11>은 프로세스의 순차 제어 구조가 제어 구조의 목록을 구성 요소로 가지는 것을 표현한 것이다. 위의 표현법에서 P는 Components, X

는 Sequence, C는 ControlConstructList를 나타낸다.

● 변환 규칙 3.3 온톨로지 매핑

온톨로지 매핑은 기존의 온톨로지를 다른 새로운 온톨로지에 적용하기 위해 클래스나 속성을 재 정의한다. 만일 P(X, Y)로 표현될 경우, X는 Y와 같은 클래스나 속성으로 정의된다. OWL-S는 equivalentClass와 equivalentProperty를 이용하여 이전 버전에서

```

OWL-S :
<owl:Class rdf:ID="Sequence">
...
<owl:Restriction>
<owl:onProperty rdf:resource="#components" />
<owl:allValuesFrom
rdf:resource="#ControlConstructList" />
</owl:Restriction>
...
</owl:Class>

TM-S :
<association>
<instanceOf>
<topicRef xlink:href="#Process.xtr#components"/>
</instanceOf>
<member> <roleSpec>
<topicRef xlink:href="#structure"/>
</roleSpec>
<topicRef xlink:href="#Sequence"/>
</member>
<member> <roleSpec>
<topicRef xlink:href="#allValuesFrom"/>
</roleSpec>
<topicRef xlink:href="#ControlConstructList"/>
</member>
</association>
    
```

<그림 11> OWL-S에서 순차 구조에 대한 구성 요소 정의

사용된 클래스와 속성을 재 정의하고 있다. 이 표현은 규칙 3을 그대로 적용하여 Topic Maps에서 P(Rx: X, Ry: Y)로 표현된다. 이 규칙은 OWL-S에서 이전 버전에 정의된 ProcessComponent 클래스를 ControlConstruct 클래스로 재 정의할 때 사용한다.

다음 변환 규칙 4와 규칙5는 Topic Maps의 다중 관계 특성과 다방향 관계 특성을 이용하여 개념들간의 관계를 정의한다.

● 변환 규칙 4 집합 연산자

집합 연산자는 이미 정의된 클래스들을 이용하여 새로운 클래스를 표현한다. 예를 들어, Y_1, Y_2, \dots, Y_n 클래스들을 모두 X 클래스로 정의할 때 집합 연산자를 사용한다. OWL은 합집합(UnionOf), 여집합(ComplementOf), 교집합(IntersectionOf), 열거형 클래스(oneOf), 분리 클래스(disjointWith)와 같은 연산자들을 제공한다.

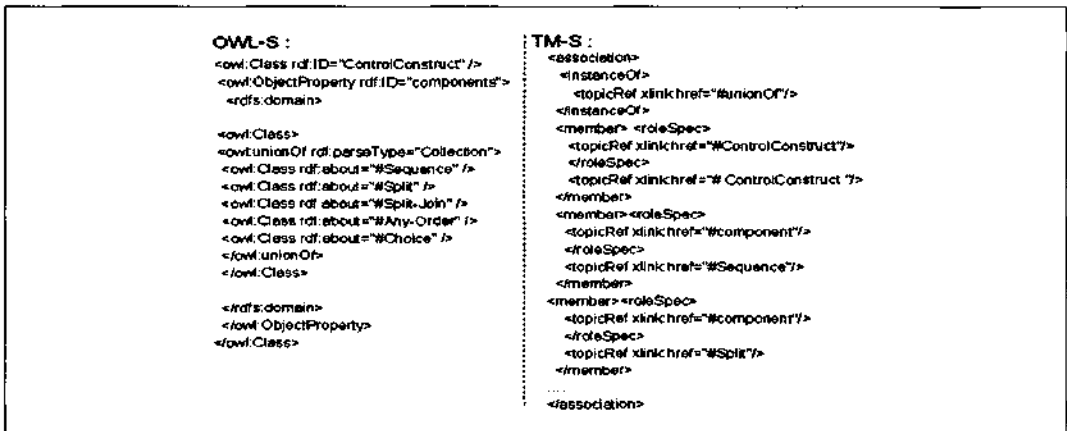
$$OWL: X = P(Y_1, Y_2, \dots, Y_n)$$

$$\equiv P(Rx: X, \text{component: } Y_1, \text{component} : Y_2, \dots, Y_n): \text{Topic Maps}$$

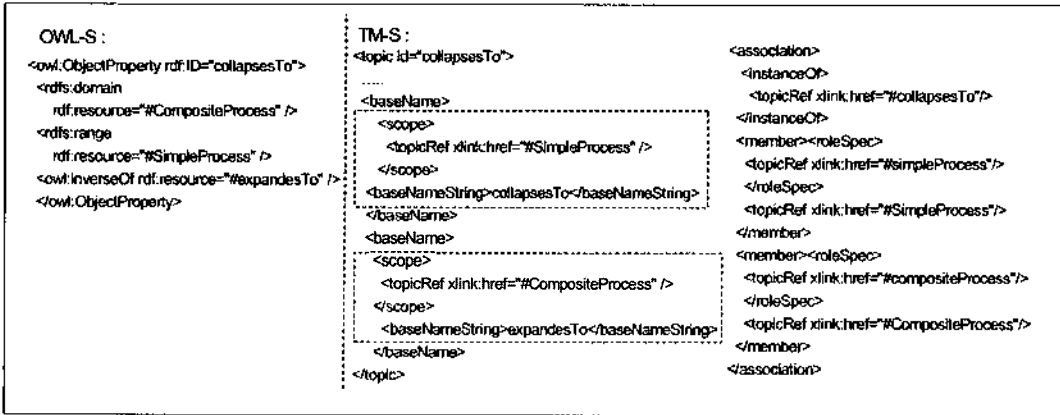
Topic Maps는 다중 관계에 대한 정의가 가능하다. Association에는 1개 이상의 Topic들이 참가하여 관계를 정의할 수 있다. 또한, 이 Topic들은 서로 다른 역할을 가지고 관계에 참여하는 것이 가능하다. Topic Maps는 집합 연산자에서 구성 요소의 역할을 하는 Topic Y_1, Y_2, \dots, Y_n 들을 이용하여 새로운 Topic X를 정의한다. OWL-S는 프로세스의 종류, 제어 구조의 구성 정보, input/output 간의 관계를 정의하기 위해 이 연산자를 사용한다. <그림 12>는 제어 구조를 순차, 분리, 분리+결합과 같은 구조들로 구성됨을 나타낸다.

● 변환 규칙 5 역 관계

OWL에서 inverseOf 속성은 속성에 대한 추론을 제공하는 방법 중에 하나이며, 속성의 역을 정의한다. 만일 P2가 P1에 대한 역



<그림 12> 제어 구조의 구성 요소



〈그림 13〉 단순 프로세스와 복합 프로세스간의 관계 표현

관계일 경우, 모든 X, Y에 대해 P1(X, Y) 이면 P2(Y, X) 가 성립되는 것을 추론할 수 있다. OWL에서는 단 방향 관계 정의만 가능하므로 다음과 같이 inverseOf를 이용하여 P1에 대한 역인 P2를 정의해야 한다.

$$OWL : P1(X, Y) \wedge inverseOf(P1, P2)$$

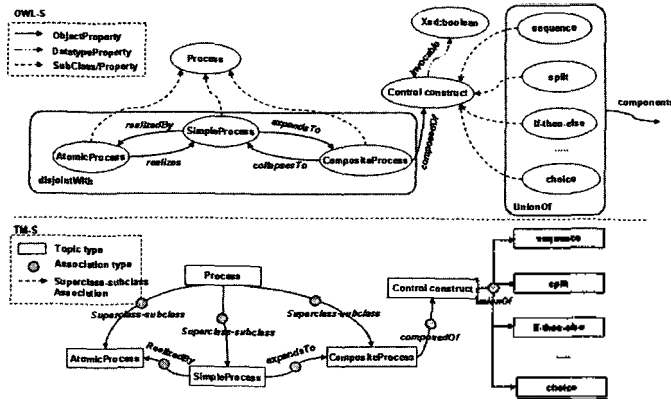
Topic Maps는 다방향 관계에 대한 정의가 가능하다. 이것은 Association에 참여하는 Topic의 역할을 이용하여 정의한다. P1(Rx: X, Ry: Y)이 Rx에서 Ry로의 관계를 가진다면, 역관계는 Ry에서 Rx로의 관계를 의미한다. 관계 명칭은 Association P1을 정의한 Topic P1의 Basename에 정의한다. Topic P1의 <scope>이 Rx인 <basename>에 P1을 정의한다. Topic P1의 <scope>이 Ry인 <basename>에 P2를 정의한다. 〈그림 13〉은 단순 프로세스와 복합 프로세스간의 관계를 정의한 것이다. 단순 프로세스는 복합 프로세스로 확장되어 표현되며, 복합 프로세스는 단순 프로세스로 축약되어 표현된다.

지금까지의 규칙을 적용하여 〈그림 14〉와 같이 OWL-S 프로세스 모델을 Topic Maps 기반의 프로세스 구조 온톨로지로 변환한다.

4.2 프로세스 개념 온톨로지 표현

프로세스 개념 온톨로지는 개념들간의 계층 관계와 개념에 대한 다양한 관점을 가진다. Topic Maps의 Superclass-Subclass 타입의 Association은 개념들간의 계층 관계 및 다중 관계를 표현할 수 있다. 그리고 Association의 Scope은 개념들간의 관계에 대한 관점들을 정의할 수 있다. 〈그림 15〉는 판매 개념에 대해 Topic Maps 모델로 표현한 것이다.

〈그림 16〉은 〈그림 15〉를 XTM으로 표현한 것이다. 이 예에서 판매 개념을 정의하는 'sell', 'sell via store', 'sell via physical store', 'sell via electronic store' 등은 Topic으로 표현된다. 'sell'와 'sell via store'간의 계층 관계는 superclass-Subclass 타입의 Association으로 표현된다. 이때, 상위 개념인 Topic 'sell'은 superclass 역할을 하며, 하위 개념인 Topic



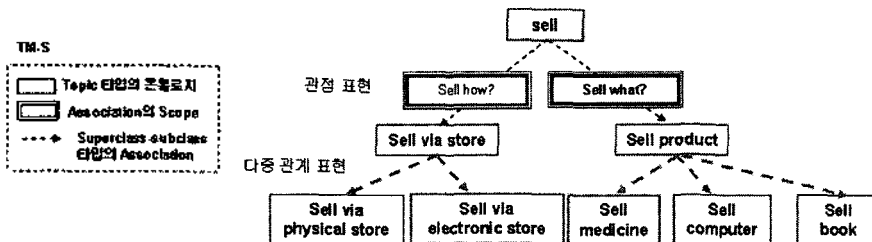
〈그림 14〉 OWL-S 프로세스 모델과 TM-S의 프로세스 구조 모델

‘sell via store’는 subclass 역할을 한다. 개념 간의 다중 관계를 가지는 ‘sell via store’와 ‘sell via physical store/sell via electronic store’ 간의 관계도 하나의 Association으로 표현된다. 그리고 ‘sell’과 ‘sell via store’ 간의 관계에 대한 관점은 ‘sell how’ 값을 가지는 Scope에 표현된다.

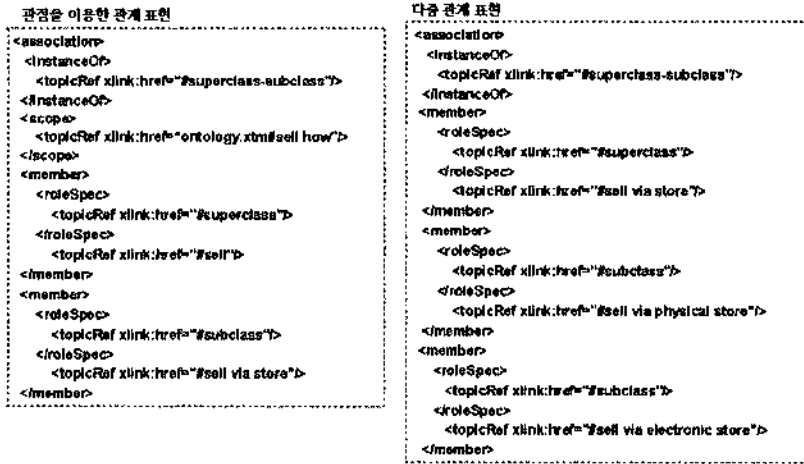
4.3 프로세스와 프로세스 개념 온톨로지와의 관계 표현

모든 프로세스는 프로세스 개념 온톨로지에 정의된 한 개 이상의 개념들을 이용하여

프로세스의 의미를 표현할 수 있다. 〈그림 17〉은 전자 상점에서 도서를 판매하는 서비스에 대한 프로세스 구조 온톨로지와 이 온톨로지에서 프로세스 개념 온톨로지를 표현한 것이다. 도서 판매 서비스는 복합 프로세스인 도서 판매 프로세스로 표현된다. 이 프로세스는 여러 개의 원자 프로세스들과 복합 프로세스들로 구성되어 있다. 도서 판매 프로세스는 판매에 관한 것이므로, ‘sell’ 개념을 이용한다. 이 프로세스는 판매 방법(‘sell how’)과 판매 종류(‘sell what’)에 대한 관점에서 의미를 부여할 수 있다. 즉, 판매 방법 측면에서는 전자 상점을 통해 판매하



〈그림 15〉 판매 개념에 대한 Topic Maps 모델



〈그림 16〉 XTM을 이용하여 판매 개념 표현

는('sell via electronic store') 의미를 가지며, 판매 종류 측면에서는 도서를 판매('sell book')하는 의미를 가진다. 원자 프로세스도 같은 방법으로 의미를 부여할 수 있다. TM-S에서 프로세스의 의미는 Topic 프로세스의 인스턴스로서 표현한다. 예를 들어, 도서 판매 프로세스는 'sell via electronic store'와 'sell book'의 의미를 가지므로, Topic 'SellBookProcess'에 <instanceOf>를 이용하여 프로세스 개념 온톨로지들을 명시한다.

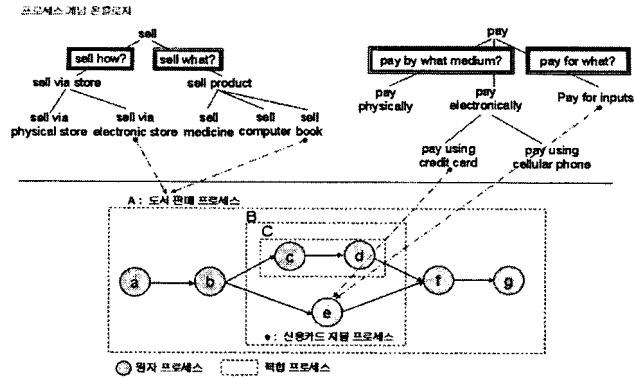
5. TM-S 기반 지능적인 발견을 위한 발견 시스템의 설계 및 구현

본 장은 사용자가 TM-S 온톨로지를 효과적으로 검색하기 위해, 검색 요소들과 요소들 간의 관계를 정의하는 클래스 다이어그램을 설명한다. 또한, 이 클래스 다이어그램

을 기반으로 사용자가 직접 질의를 할 수 있도록 질의어를 정의한다. 그리고 22절에서 기존 연구의 문제점을 지적하기 위해 사용된 질의 예들을 본 장에서 정의한 질의어를 이용하여 표현함으로써, 사용자 입장에서 프로세스 정보를 표현한 온톨로지를 어떻게 검색하는지를 보여준다.

5.1 질의 클래스 다이어그램

TM-S 온톨로지 기반의 질의 클래스 다이어그램은 〈그림 19〉와 같이 3가지의 클래스 즉, 개념 클래스(concept class), 프로세스 클래스(process class), input/output 클래스로 정의되어 있다. 개념 클래스는 프로세스 개념 온톨로지에 정의된 정보를 검색하는 것으로, 개념의 이름과 분류한 관점, 그리고 개념들 간의 구조 검색을 제공한다. 프로세스 클래스는 프로세스 시그니처 온톨로지과 프로세스 구조 온톨로지에 대한 검색을 제공한다.



〈그림 17〉 프로세스와 프로세스 개념 온톨로지간의 관계

```

<topic id="SellBookProcess">
  <instanceOf>
    <topicRef xlink:href="Ontology.xtm#SellViaElectronicStore" />
  </instanceOf>
  <instanceOf>
    <topicRef xlink:href="Ontology.xtm#SellComputerBook" />
  </instanceOf>
  ...
  <baseName>
    <baseNameString>Sell Book Process</baseNameString>
  </baseName>
</topic>
    
```

프로세스의 의미 표현

〈그림 18〉 XTM을 이용하여 Topic 'SellBookProcess'의 의미 표현

이 클래스는 프로세스 시그니처 온톨로지에 정의된 프로세스 이름을 검색하고, 프로세스 구조 온톨로지에 정의된 구조 정보를 검색하기 위해 여러 가지 함수들을 제공한다. input/output 클래스는 프로세스 시그니처 온톨로지에 정의된 input/output 이름을 검색한다. 그리고 개념 클래스는 프로세스 클래스와의 관계를 통해 특정 개념에 포함되는 프로세스들을 검색할 수 있다. 프로세스 클래스는 input/output 클래스를 이용하여 프로세스에 관련된 input/output들을 검색한다.

5.2 확장된 질의 표현

사용자는 TMS의 세부적인 온톨로지 구조에 대한 지식이 없으므로, 프로세스 정보들을 쉽게 검색할 수 있는 방법이 필요하다. 본 논문에서는 5.1절에서 정의한 클래스 다이어그램을 기반으로 검색할 수 있는 TMS-QL을 정의하였다. TMS-QL은 OQL(Object Query Language) 표현법을 기반으로 하며, TM-S 온톨로지에 정의된 정보를 검색하기 위해 다음과 같이 표현법을 확장하였다.

정의 1: 개념에 대한 관점 정의는 다음과 같이 표현된다.

{클래스 | 애트리뷰트} ^ [Scope]

TM-S 온톨로지에서 Scope은 프로세스 개념 온톨로지에서 분류에 대한 관점을 명시하거나, 프로세스나 input/output이름에 대한 표현 범위를 정의하는데 사용된다. {클래스 애트리뷰트} ^ [scope]는 scope에 포함되는 클래스 또는 애트리뷰트들을 의미한다. 예를 들어, Ontology ^ [sell what]은 'sell what' 분류에 속하는 개념 클래스들을 말한다. Process.name ^ [Korean] 표현은 한국어로 정의된 프로세스의 이름은 나타낸다.

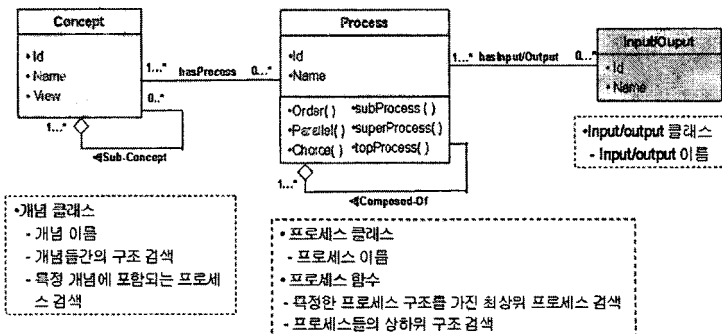
정의 2: 프로세스 구조를 검색하기 위해 6 가지 함수를 이용한다.

프로세스 모델은 프로세스 구조를 표현하기 위해 7가지 제어 구조를 제공한다. 즉, 순차(Sequence), 분리(Split), 분리+결합(Split+Join), 선택(Choice), 임의-순서(any-

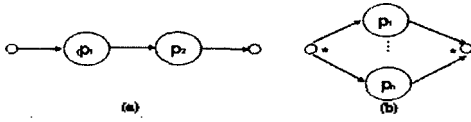
order), 양자택일(if-then-else), 반복(iterate) 이다. 프로세스 구조는 이 제어 구조들을 이용하여 복잡한 형태로 정의된다. 그러나 대부분의 사용자는 단순한 형태의 프로세스 구조를 질의한다. 본 논문에서는 사용자의 질의에 유용하도록 제어 구조를 3가지로 축소하여 프로세스 구조를 검색할 수 있는 함수를 정의하였다.

(1) Order (): 이 함수는 하위 프로세스들이 순차적으로 처리되는 프로세스를 검색한다. 즉, "배송 후, 결제가 가능한 도서 판매 프로세스를 찾아라." 와 같은 질의에서 하위 프로세스인 배송 프로세스(P₁)와 결제 프로세스(P₂)가 <그림 20>의 (a)와 같이 순차적으로 처리되는 도서 구매 프로세스들을 검색한다. 이 함수는 제어 구조들 중에서 순차와 임의-순서 구조를 가지는 프로세스들을 분석한다.

(2) Parallel (): 동일 프로세스에서 분기된 하위 프로세스들이 병렬적으로 처리되는 프로세스를 검색한다. <그림 20>의 (b)와 같이, "배송(P₁)과 재고품 갱신



<그림 19> TM-S의 질의 클래스 다이어그램



〈그림 20〉 순차 구조(a)와 병렬 구조(b) 모델

(P_2)이 병렬적으로 수행되는 도서 판매 프로세스를 찾아라.”와 같은 질의를 처리하기 위해 사용된다. 이 함수는 프로세스 제어 구조 중에 분리와 분리조합 구조로 정의된 프로세스들을 검색한다.

- (3) Choice () : 이 함수는 2개 이상의 프로세스들 중에서 조건에 맞는 하나의 프로세스만 선택하여 처리하는 구조를 가진 프로세스들을 검색한다. 즉, “신용카드 결제(P_1) 또는 무통장 입금 결제(P_2)가 가능한 도서 판매 프로세스를 찾아라.”와 같은 질의 처리에 사용된다. 이 함수는 선택 구조를 제공하는 선택과 양자택일 구조를 가지는 프로세스들을 분석하여 검색한다.

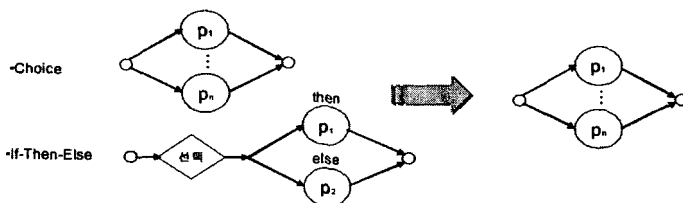
제어 구조들 중에서 반복은 하나 또는 그 이상의 프로세스들이 반복되어 처리되는 구조이다. 이 구조는 일반적으로 사용자의 질

의에서 자주 사용되지 않는다. 그러므로 〈그림 22〉과 같이 반복 구조는 하나의 동작을 하는 프로세스로 취급하여 프로세스 구조를 분석하였다. 본 논문에서는 제어 구조를 이용한 검색 외에, 특정 프로세스의 상위 프로세스들을 검색하는 superProcess(), 특정 프로세스의 하위 프로세스들을 검색하는 subProcess(), 최상위 프로세스인 서비스를 검색하는 topProcess() 함수들을 제공한다.

5.3 질의 예

본 논문의 22절에서 언급한 질의에 대해 TMS-QL로 질의를 정의하여 사용자가 프로세스 정보를 표현한 온톨로지를 검색하는 방법을 보여준다.

- “전화번호로 결제할 수 있는 도서 판매 서비스를 찾아라.”(프로세스와 input/output을 이용한 질의): 이 질의는 검색을 위한 조건을 정의하기 위해, 프로세스 클래스와 input/output 클래스를 이용하여 프로세스 이름과 input 이름을 정의한다. 그리고 함수 topProcess()를



〈그림 21〉 선택 구조의 모델



〈그림 22〉 반복 구조의 모델

이용하여 최상위 프로세스의 이름이 '도서 판매 서비스'를 포함하는 것을 찾는다.

```
Select p.topProcess( )
From Process p, phasInput i
Where p.name contains '결제' and i.name
contains '전화 번호' and p.topProcess like
'%도서 판매 서비스 %'
```

- “배송 후 결제를 제공하는 도서 판매 서비스를 찾아라.” (프로세스 구조를 이용한 질의): 이 질의는 Order() 함수를 이용하여 배송과 결제가 순차적으로 처리되는 최상위 프로세스를 검색한다.

```
Select p.name
From Process p
Where p.Order( '배송', '결제' ) and p.name
like '%도서 판매 서비스%'
```

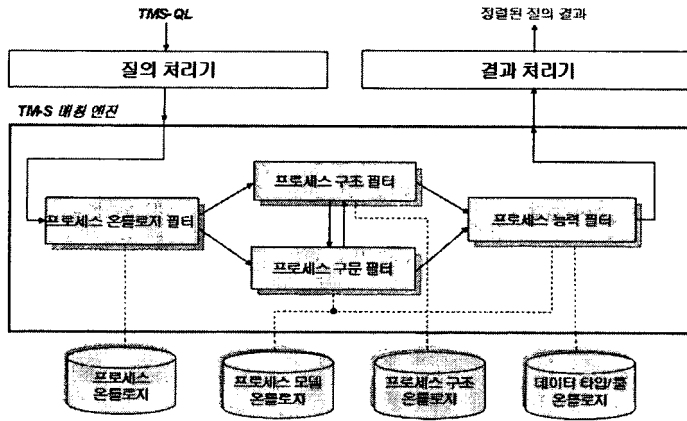
- “전자 상점을 통해 고객에게 책을 판매하는 프로세스를 찾아라.”(프로세스 개념을 이용한 질의) : 이 질의는 판매 방법('sell how')으로 '전자 상점', 판매 대상('sell who')으로 '고객', 판매 종류('sell what')으로 '책을 판매'하는 개념에 모두 속하는 프로세스들을 찾는다.

```
Select p.name
From Ontology o, o.hasprocess p
Where o[ 'sell how' ].name like '%
```

전자 상점%' and o['sell who'].name like '%고객%' and o['sell what'].name like '%책 판매%';

5.4 시스템 구현

본 시스템의 구성 요소들은 자바(J2SDK 1.4.1_02)를 기반으로 구현되었다. TM-S 온톨로지를 검색하기 위해 온토피아에서 개발한 OKS(Ontopia Knowledge Suite) 엔진은 사용하였다[16]. 이 엔진은 데이터베이스에 저장되어 있는 XTM 파일에 대해 Topic, Association, Occurrence의 값들을 추출해 오는 기능을 한다. 본 논문과 관련된 전체 시스템 구조는 〈그림 23〉과 같다. 사용자 인터페이스를 통해 직접 TMS-QL 형태의 질의를 입력받거나, 필-아웃 폼 형태의 인터페이스를 통해 질의 정보를 입력 받은 후 TMS-QL 형태의 질의를 생성된다. 질의 처리기는 입력 받은 질의를 분석 한 후, 매칭 엔진을 통해 처리할 실행 계획을 세운다. 질의 처리기에서 이용한 TMS-QL 형태의 질의어 분석 및 TMS-QL 질의 대수에 대한 정의는 본 논문과 관련된 이전 연구 결과에서 수행된 것으로, 상세한 내용은 이전의 발표된 여러 논문지를 통해 참고할 수 있다[2,3]. 매칭 엔진은 실행 계획에 따라 각 필터들을 이용하여 온톨로지 정보들을 검색한다. 매칭 엔진을 통해 검색된 결과는 결과 처리기를 통해 매칭 유사도와 매칭 요소에 대한 가중치를 고려하여 결과에 대한 순위를 계산한 후 사용자에게 보여준다. 매칭 엔진을 이용한 시스템 구현은 본



〈그림 23〉 TM-S 시스템의 구조

논문의 기술 범위에서 벗어나며, 이전 논문지에 상세히 설명하였다[1].

6. 결 론

본 논문은 기존의 시맨틱 웹 서비스 발견의 문제점들을 분석하였으며, 이를 해결하기 위해 프로세스 정보를 온톨로지로 표현한 TM-S 온톨로지와 질의어인 TMS-QL를 제안하였다. 이 온톨로지는 3가지의 하위 온톨로지로 나누어 정의하였다. 즉, 프로세스의 텍스트 정보를 정의한 프로세스 시그니처 온톨로지, 프로세스의 구조 정보를 정의한 프로세스 구조 온톨로지, 프로세스의 의미를 정의하는 프로세스 개념 온톨로지이다. 이 온톨로지들의 지식 구조는 Topic Maps를 이용하여 체계적으로 표현하였다. 또한, TMS-QL을 이용한 질의 예를 보여줌으로써 기존의 발견이 가지는 문제점들을 해결하는 구체적인 모습을 보여주었다.

본 논문에서 프로세스 시그니처 온톨로지와 프로세스 구조 온톨로지는 서비스 발견의 정확률을 향상시켰으며, 프로세스 개념 온톨로지는 발견의 재현율을 높이는 효과를 제공하였다. 또한, 프로세스 정보를 표현한 온톨로지 정의와 사용자가 직접 질의 할 수 있는 질의어는 웹 서비스의 발견을 실제적으로 가능케 하는 구체적인 방법을 제공하였다. 이 연구 결과는 웹 서비스를 위한 온톨로지 모델링 연구에 활용될 수 있다. 또한, 서비스의 자동화를 위한 발견 시스템은 물론, 서비스의 조합, 통합 및 상호 운영을 위한 시스템에 기본이 되는 연구로 활용될 수 있다. 본 논문에서 제안한 연구가 기존의 관련된 연구들보다 정확하고 관련된 서비스 검색 결과를 제공한다는 것은 본 논문에서 해결한 질의 예들을 통해 직관적으로 알 수 있다. 현재, OWL-S, WSMO, SWSO 등의 온톨로지 정의가 아직 진행 중에 있으며, 이를 기반으로 한 인스턴스에 대한 정보가 미흡하다. 이로 인해, 검색의 정확성 및 데이터

의 처리 시간에 대한 성능 분석에 한계점을 가지고 있는 실정이다. 그러므로 향후 연구로써 온톨로지 및 인스턴스에 대한 자료가 확보되면 이를 이용한 성능 분석이 진행될 예정이다.

참 고 문 헌

- [1] 유소연, 유정연, 이규철, "프로세스 정보와 QoS를 고려한 웹 서비스 발견," 한국전자거래학회지, 제10권, 제3호, pp. 85-110, 2005
- [2] 황윤영, "Topic Maps 기반의 지능적 웹 서비스 시스템 설계 및 구현," 충남대학교 석사 학위논문, 2004.
- [3] 황윤영, 유정연, 유소연, 이규철, "지능적 웹 서비스 발견 시스템에서의 질의 처리," KDBC 2004 학술발표논문집, 한국정보과학회 데이터베이스연구회, pp. 11-18, 2004.
- [4] Berners-Lee, T. et al., "The Semantic Web," Scientific American, pp. 34-43, 2001.
- [5] Bernstein, A and Klein, M., "Towards High-Precision Service Retrieval," International Semantic Web Conference 2002, Springer-Verlag, pp. 84-101, 2002.
- [6] Booth, D. et al., Web Services Architecture, W3C Working Group Note, 2004, <http://www.w3.org/TR/ws-arch/>.
- [7] Cardoso, J., and Sheth, A., "Semantic e-workflow composition," Journal of Intelligent Information Systems, Vol. 21, No. 3, pp. 191-225, 2003.
- [8] Gomez-Perez, A and Corcho, O., "Ontology Specification Languages for the Semantic Web," IEEE Intelligent Systems Vol. 17, No. 1, pp. 54-60, 2002.
- [9] Gruber, T., "A translation approach to portable ontologies," Knowledge Acquisition, Vol. 5, No. 2, pp. 199-220, 1993.
- [10] Keller, U. et al., (Ed.), D5.1v0.1 WSMO Web Service Discovery, WSMO Working Draft, 2004. <http://www.wsmo.org/TR/d5/d5.1/v0.1/>
- [11] Martin, D. et al., OWL-S: Semantic Markup for Web Services, 2004. <http://www.daml.org/services/owl-s/1.1/overview/>
- [12] McGuinness, D. L, and Harmelen, F.van., Editors, OWL Web Ontology Language Overview, W3C Recommendation, 2004. <http://www.w3.org/TR/owl-features/>
- [13] McIlraith, S.A. and Martin, D.L., "Bringing Semantics to Web Services," IEEE Intelligent Systems, Vol. 18, No. 1, pp. 90-93, 2003.
- [14] McIlraith, S.A. et al., "Semantic Web Services," IEEE Intelligent Systems, Vol. 16, No. 2, pp. 46-53, 2001.
- [15] Obrst, L., "Ontologies for semantically interoperable systems," In Proceedings of the 2003 ACM CIKM International Conference on Information and Know-

- ledge Management, pp. 366-369, 2003.
- [16] Ontopia, "The Ontopia Knowledge Suite".
<http://www.ontopia.net/ontopia/texts/product-wp.html>, 2003.
- [17] Paolucci, M., et al., "Semantic Matching of Web Services Capabilities," The First International Semantic Web Conference, Springer-Verlag, pp. 333-347, 2002.
- [18] Pepper, S., "The TAO of Topic Maps: finding the way in the age of infoglut," XML Europe, 2002.
- [19] Pepper, S. and Moore, G., XML Topic Maps (XTM) 1.0, TopicMaps.Org Specification, 2001.
<http://www.topicmaps.org/xtm/1.0/>.
- [20] Roman, D. et al., Editors, D2v12 Web Service Modeling Ontology (WSMO), WSMO Final Draft, 2005.
<http://www.wsmo.org/TR/d2/v1.2/>.
- [21] Semantic Web Services Initiative, <http://www.swsi.org/>.

저 자 소 개



유정연
1999. 2.
2006. 8.
현재
관심분야

(E-mail : jyyou@cnu.ac.kr)
충남대학교 대학원 컴퓨터공학과 (공학석사)
충남대학교 대학원 컴퓨터공학과 (공학박사)
충남대학교 소프트웨어연구센터 전임연구원
데이터베이스, XML, 시맨틱 웹 서비스



이규철
1984. 2.
1986. 2.
1990. 8.
1994.
1995 ~ 1996.
2001 ~ 현재

2003 ~ 현재
2003 ~ 현재
현재
관심분야

(E-mail : kdee@cnu.ac.kr)
서울대학교 공과대학 컴퓨터공학과 (공학사)
서울대학교 공과대학 컴퓨터공학과 (공학석사)
서울대학교 공과대학 컴퓨터공학과 (공학박사)
미국 IBM Almaden Research Center 초빙 연구원
미국 Syracuse University 초빙 교수
전자상거래 표준화 통합 포럼 전자거래 기반
기술위원회 위원장
한국전자거래학회 편집이사
웹 코리아 포럼 부위원장
충남대학교 공과대학 컴퓨터공학과 교수
XML, 웹 서비스, 시맨틱 웹 서비스, 유비쿼터스 웹 서비스