

# 시맨틱 웹 기반 상황인지 서비스를 위한 동적 서비스 제공 모델

(A Dynamic Service Supporting Model for Semantic  
Web-based Situation Awareness Service)

최정화<sup>†</sup>      박영택<sup>\*\*</sup>

(Jung-Hwa Choi)      (Young-Tack Park)

**요약** 시맨틱 웹 서비스 기술은 상황 인지의 실현을 위한 기반 기술로 다양한 자원들을 동적이고 유연하게 상호 융합하여 새로운 서비스를 생성한다. 유비쿼터스 컴퓨팅 기술이 현실화됨에 따라 웹 서비스를 구현하려는 연구가 활발하지만, 대부분이 웹 서비스 설계자의 최초 명세에 국한된 획일화된 서비스 결과만을 초래한다. 본 논문에서는 사용자 요구와 감지한 상황의 월드 모델을 분석하여 계획 시스템에 목표와 초기 상태로 입력하고 초기 상태에서부터 목표를 달성하기 위한 일련의 작업들을 계획하는 동적 서비스 제공을 위한 모델링 방법을 제안한다. 제안한 방법론은 실제계로부터 감지한 월드모델을 OWL 도메인 온톨로지를 이용하여 서술논리 기반 온톨로지 추론을 통해 상황정보(context)를 추론한다. 상황정보는 서비스 도메인을 결정하며, 이에 해당하는 OWL-S 서비스 온톨로지를 계획 시스템에서 탐색할 서비스 명세로 활용한다. 계획 시스템은 초기 상태에서 목표 상태를 만족하는 하나 이상의 서비스를 탐색하고 실행 순서를 계획한다. 이 시스템은 STRIPS 형의 역방향 탐색 시스템으로 OWL-S 서비스를 AI 전통 계획 방법론에 근거하여 합성하여, 방대한 웹 서비스의 탐색 범위를 축소한다. 또한 패턴 매칭에 의해 실행 가능한 서비스를 찾지 못한 경우, DL기반의 시맨틱 매칭을 통해 대안이 되는 서비스를 찾는다. 제안한 방법은 비교연구인 OWLS-XPlan과 동일한 시나리오로 실험하여 기존 연구의 문제점을 해결하고, 동적 서비스 실현을 위한 모델링 방법으로써 가능성을 검증한다.

**키워드** : 웹 서비스, 계획, 서비스 합성, 온톨로지, 상황인지, 상황정보(컨텍스트)

**Abstract** The technology of Semantic Web realizes the base technology for context-awareness that creates new services by dynamically and flexibly combining various resources (people, concepts, etc). According to the realization of ubiquitous computing technology, many researchers are currently working for the embodiment of web service. However, most studies of them bring about the only predefined results those are limited to the initial description by service designer. In this paper, we propose a new service supporting model to provide an automatic method for plan related tasks which achieve goal state from initial state. The inputs on an planner are intial and goal descriptions which are mapped to the current situation and to the user request respectively. The idea of the method is to infer context from world model by DL-based ontology reasoning using OWL domain ontology. The context guide services to be loaded into planner. Then, the planner searches and plans at least one service to satisfy the goal state from initial state. This is STRIPS-style backward planner, and combine OWL-S services based on AI planning theory that enabling reduced search scope of huge web-service space. Also, when feasible service do not find using pattern matching, we give user

· 본 연구는 숭실대학교의 지원을 받았습니다.

† 학생회원 : 숭실대학교 컴퓨터학과  
cjh79@ailab.ssu.ac.kr

\*\* 종신회원 : 숭실대학교 컴퓨터학과 교수  
park@ssu.ac.kr

논문접수 : 2009년 4월 6일

심사완료 : 2009년 7월 17일

Copyright©2009 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 소프트웨어 및 응용 제36권 제9호(2009.9)

alternative services through DL-based semantic searching. The experimental result demonstrates a new possibility for realizing dynamic service modeler, compared to OWLS-XPlan, which has been known as an effective application for service composition.

**Key words** : Web Service, Planning, Service Composition, Ontology, Situation Awareness, Context

## 1. 서론

시맨틱 웹(Semantic Web) 서비스는 웹상에서 유통되는 데이터들에 대한 온톨로지(ontology)를 구축함으로써 기계가 이해 가능한 데이터 집합을 이용하여 자동화된 데이터 처리를 수행한다. 유비쿼터스 컴퓨팅 기술이 현실화됨에 따라 사용자에게 언제 어디서나 맞춤형된 서비스를 제공하기 위한 연구가 확대되고 있으며, 사용자의 상황(situation)과 서비스 시맨틱 기술(description)을 기반으로 적절한 서비스 혹은 세부 단위 컴포넌트를 동적으로 조합하여 제공하는 방법론이 요구된다. 대부분의 선행 연구들은 서비스와 기기를 명확하게 분리된 개념으로 다루지 않고 있으며 기기 측면에 더 초점을 맞추고 있다. 즉 서비스 보다는 그 서비스를 제공할 수 있는 기기의 분류, 구조화, 대체를 통해 동적 지원과 가용성 향상을 이루려 해왔다. 앞으로의 전망은 구조화된 서비스의 사용을 위해 서비스 온톨로지를 사용하여 사용자의 목적 달성 성공률을 높이는 것과 실시간의 상황을 고려한 목적 달성에 필요한 서비스들의 자동화된 합성(composition) 기법[1,2]의 연구가 요구된다. 따라서 웹 서비스의 등록과 발견을 위해 UDDI(Universal Description, Discovery, and Integration)<sup>1)</sup> 기술이 아닌 시맨틱 웹 서비스에서의 온톨로지 상위 클래스 간의 관계성 등을 이용하여 사용자 요구에 자동화된 의미 검색을 지원하는 연구가 필요하다[3-5].

본 연구에서는 사용자 요구와 감지(sensing)된 상황의 월드 모델을 분석하여 수행할 서비스 목표(goal)와 초기(initial) 상태로 설정하고 초기 상태에서부터 목표를 달성하기 위한 일련의 작업(들)을 계획하는 동적 서비스 제공을 위한 모델링 방법을 제안한다. 이 방법론은 서술논리(Description Logic; DL)기반의 온톨로지 추론기술을 이용하여 상황을 인지하고, OWL-S로 정의된 서비스들 중 사용자 요구와 상황에 맞는 서비스들을 패턴매칭과 시맨틱(semantic) 매칭 기술에 의해 발견하고 합성하는 계획 시스템(planner)을 포함한다. 제안한 계획 시스템은 STRIPS[6] 형의 역방향(backward) 탐색 알고리즘을 이용한다.

제안하는 방법은 온톨로지에 정의된 지식을 이용함으로써 찾고자 하는 명세에 완전히 일치하는 서비스가 정의되어 있지 않더라도 유사한 서비스의 탐색이 가능한

장점을 가지고 있다. 완전한(complete) 추론엔진의 구축이 힘들듯이 이 기술은 완전할 수 없는 서비스 저장소의 구축의 단점을 보완해준다. 제안한 모델은 OWL 온톨로지[7], OWL-S 온톨로지[8], 그리고 STRIPS 기반 계획 시스템의 지식 베이스를 정보 표현의 의미적 요소가 깨지지 않게 상호 작용하여 동적 서비스 제공 모델의 가능성을 보여준다.

본 논문은 다음과 같은 순서로 구성된다. 2장에서는 온톨로지 기반 동적 서비스 모델링을 위한 관련 기술들을 살펴보고 3장에서는 시맨틱 웹 기반의 동적 서비스 모델을 정의하고, 4장에서 이 모델에 상황인지를 결합한 동적 서비스 제공 모델의 구조를 컴포넌트 별로 설명한다. 5장에서는 제안한 방법의 타당성과 정확성을 검증하기 위해 기존 연구 OWLS-XPlan과 비교한 실험결과를 기술한다. 마지막 6장에서는 결론을 맺고 향후 연구를 제시한다.

## 2. 관련 연구

### 2.1 OWL 기반 웹 서비스 온톨로지, OWL-S

서비스의 상위 온톨로지를 제공하는 OWL-S[8]는 기존의 웹 서비스의 표준화된 기술을 대체하는 것이 목적이 아니라 시맨틱 계층을 제공하는 것에 있다. OWL-S의 웹 서비스 호출은 WSDL[9]에 근간을 두고 있으며, 웹 서비스의 발견 역시 UDDI를 확장하는 방식으로 접근한다. 즉, OWL-S 온톨로지로는 웹 서비스를 기술하여 자동으로 서비스를 발견하고, 합성하고, 실행하는 것을 지원하도록 디자인 되어있다. OWL-S의 시맨틱 계층은 Service 클래스와 속성으로 연결된 다음의 3개의 하위 클래스로 구성된다.

- 서비스 프로파일(profile) : 사용자가 요구하는 서비스가 무엇(what)인지를 정의한다.
- 서비스 모델(model) 또는 프로세스 : 서비스를 어떻게 작동하는지를 명세한다.
- 서비스 그라운드링(grounding) : 어떻게 서비스를 사용하는지에 대한 정보를 준다.

하나의 서비스는 Service 클래스의 인스턴스로 정의되고, 속성으로 위의 세 개의 클래스의 인스턴스를 연결한다. 속성 “presents”는 프로파일, “describedBy”는 모델, “supports”는 그라운드링과 연결된다. 서비스 모델은 서비스를 실행할 때 필요한 입력(Inputs), 서비스를 실행한 후에 얻게 되는 출력(Outputs), 서비스를 실행하기

1) <http://www.uddi.org>

위해 만족되어야 하는 전제조건(Preconditions), 그리고 서비스를 실행한 후에 발생하는 효과(Effects)의 *IOPE* 용어(term)를 사용하여 서비스를 정의한다. 이 서비스 모델은 복합 서비스를 정의 할 수는 있지만, 합성을 수반하는 서비스들을 자동으로 찾는 것을 지원하지 못한다. OWL-S는 단지 서비스들의 수행 계획이 정의되었을 때 유용하다. 본 논문에서는 OWL-S로 서비스를 명세하고, 온톨로지 기반 상황인지 기술을 활용하여 자동화된 서비스 경로를 계획한다. 이 방법은 *IOPE*가 매칭되는 서비스를 찾지 못했을 때 대안이 되는 서비스의 검색을 가능하게 한다.

## 2.2 웹 서비스 합성(Composition)에 관한 연구

OWLS-XPlan[1]은 OWL-S 기반 웹 서비스 컴포지션 시스템으로써, 서비스 정보 변환기인 OWLS2PDDL를 이용하여 서비스 명세 정보인 OWL-S와 도메인 온톨로지 정보인 OWL 정보를 받아 PDDL로 변환한 후 서비스들의 실행 순서를 계획한다. 이 과정에서 OWL-S 서비스 명세 중 전제조건과 효과는 OWL-S 1.1에서 제안하는 방법과 별도로 XML 형식으로 기재하고 에이전트가 알고 있는 지식 또한 별도로 정의한다. 이는 단일화된 *IOPE*를 제공하지 못하기 때문에 실제 서비스를 정의하고 이용하는데 있어서 많은 제약과 시행착오를 요구한다. 상위 버전인 OWLS-XPlan 2.0에서는 전제조건과 효과에 대해 PDDL 형식의 명세를 OWL-S에 포함하지만, 이 역시 에이전트의 지식을 하나의 전역변수로 지정하여 전제조건과 효과를 정의하는데 계획 생성 시 전달되는 값을 고려해야하는 문제점이 있다. 또한 OWLS-XPlan은 초기와 목표(사용자 요구) 상태를 각각 OWL 온톨로지 모 델링한다. 이 방법은 실행시간에 동적으로 입력되는 목표를 처리하기에 온톨로지 스키마를 수정해야하는 문제가 발생하며, 서비스에서 입력 매개 변수로 필요한 인스턴스를 초기 상태와 목표 상태 온톨로지에 정의하는데 이 구분이 불명확하다. 이 방법은 AI 계획 방법론의 초기와 목표값 정의 이론에서 벗어난다. 본 연구는 서비스 계획 시에 초기와 목표 상태의 동적인 입력을 지원하여 상황(초기 상태)별 사용자 요구(목표 상태)에 동적으로 반응한다. 또한 별도의 파일이나 변수에 의존 하지 않도록 OWL 기반의 단일화된 *IOPE* 정의를 지원한다.

SHOP2[2]는 OWL-S 기반 웹 서비스 합성 시스템으로 HTN(Hierarchical Task Network)[10] 기반의 AI 계획 기술을 적용하여 OWL-S로 의미가 기술된 웹 서비스 합성을 수행한다. 이 연구는 초기상태의 상황 정보와 사용자 요구의 목표 상태를 가지고 서비스를 계획하는 동적인 시스템과는 달리 정적으로 초기상태와 업무(task), 즉 복합 서비스를 정의하고 업무를 수행할

세부 계획(partial ordering), 즉 명시된 단일 서비스의 순서만을 동적으로 제공한다[11]. 그러나 유비쿼터스 컴퓨팅 환경에서 운영될 서비스를 위해서는 동적인 장치 및 환경 센서 등에서 얻은 상황정보로부터 추출한 상황 지식과 사용자의 선호도 등을 이용하여 동적 서비스 검색 및 서비스 합성 기술에 대한 연구가 필요하다. 본 연구는 사용자가 존재하는 현재 상황을 동적으로 인지하여 탐색할 서비스 명세의 범위를 정한다. 현재 상황을 초기 상태로 하고, 사용자 요구를 목표 상태로 하여 추출한 서비스 명세 내에 초기 상태로부터 목표 상태를 만족하는 단일 서비스들의 순서를 계획한다. 이 때 패턴 매칭에 의해 서비스를 발견하지 못 하였을 경우, 온톨로지 추론을 이용한 시맨틱 매칭을 통해 완전할 수 없는 서비스 명세의 단점을 해결한다.

## 3. 시맨틱 웹 기반 동적 서비스 제공 모델

본 논문에서 제안하는 시맨틱 웹 기반 동적 서비스 모델은 OWL 온톨로지에 정의된 지식을 참조하여 서비스를 기술하고, 온톨로지의 의미적 상호 운용성을 이용해서 사용자 요구에 맞는 서비스 관련 제반 처리를 자동화 한다. 전통적인 방법론에서 각 단계의 부가 정보를 수동으로 입력하여 수행한 것과는 달리, 제안하는 방법론은 동적인 웹 서비스 제공 모델을 지원할 수 있다. 현재 상황의 월드 모델을 초기 상태로 하고, 사용자의 요구를 목표 상태로 입력하여, 매칭 되는 웹 서비스들을 합성을 위한 서비스들의 실행 순서를 계획한다.

### 3.1 시맨틱 웹 기반 동적 서비스 계획 모델

서비스 계획 관련 제반 기술은 서비스 발견, 선택, 그리고 합성으로 구성되며, 본 논문에서는 이 기술들을 그림 1과 같은 상호작용에 의해 한 번에 수행한다. 동적 서비스 계획 기법은 서비스 온톨로지를 구축하고 명시된 서비스가 유효하지 않다고 판단되었을 때, 온톨로지의 구조를 따라가면서 동일한 효과를 낼 수 있는 대체 서비스를 찾는다. 시맨틱 웹 서비스는 온톨로지 클래스 간의 관계성을 통해서 좀 더 자동화된 검색을 가능하게 한다. 예를 들어, "OOO 뮤직 비디오를 방 안의 텔레비전을 통해 보는 서비스를 검색"하라는 질의를 현재의 UDDI를 이용한 서비스 검색 방법은 뮤직 비디오 실행에서 제공하는 서비스 개체를 UDDI 저장소로부터 검색하고, 매칭(matching) 정보를 이용하여 해당 웹 서비스를 매칭한 이후에 파일 검색, 파일명 입력, 텔레비전 전원 입력 등의 추가적인 작업을 통해 비디오 파일을 실행해야 한다. 이 경우 필요한 서비스 모델을 지원하지 못하는 경우에는 실제 적합한 서비스의 검색이 이루어 지지 못할 수 있다. 하지만 시맨틱 웹 서비스에서는 여러 개의 서비스를 합성하기 위한 구조를 갖추고 있어서

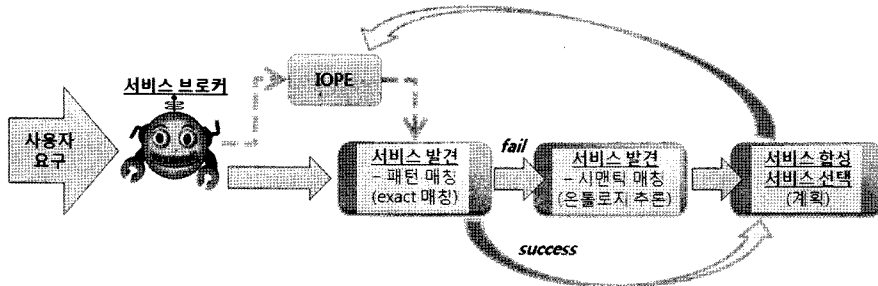


그림 1 시맨틱 웹 기반 동적 서비스 계획을 위한 흐름도

복잡한 질의도 가능하다.

그림 1의 흐름도를 통해 뮤직 비디오 예제 질의에 대한 서비스 합성 과정을 설명하면, 서비스 브로커는 사용자의 질의 입력 전 검색 시스템의 로그인 과정을 통해 수집한 '사용자 ID'와 사용자의 질의로부터의 '뮤직 비디오 제목'을 입력 받고 웹 서비스를 검색하려고 한다. 서비스 브로커는 우선 이 두 가지 정보를 가지는 IOPE 형식으로 기술된 서비스 온톨로지의 인스턴스를 검색한다. 이를 위해 서비스 브로커는 서비스 발견, 합성, 그리고 선택의 세 단계를 수행한다. 첫째, 서비스 발견은 패턴 매칭(pattern matching)과 시맨틱 매칭으로 구성된다. 우선, 패턴 매칭에 의해 정확(exact)하게 매칭 되는 인스턴스를 발견한다. 이 과정에서 해답을 얻지 못하면 서비스 브로커는 시맨틱 매칭을 수행한다. 시맨틱 매칭은 서비스 인스턴스에 있는 어휘의 의미관계를 추론하기 위해서 도메인 온톨로지 추론을 실행한다.

뮤직 비디오 제목과 사용자 ID 만으로 텔레비전에 뮤직 비디오를 틀어주는 서비스를 발견 할 수 없다면, 두 번째 기술인 서비스 합성 기술이 실행된다. 합성 과정에서 최적의 서비스들을 선정하는 것을 세 번째 기술인 서비스 선택이다. 예를 들어, 도메인 온톨로지에 뮤직 비디오 클래스의 상위 클래스로 영화가 정의되어 있다고 하자. 두 개의 입력 값에 매칭 하는 서비스를 발견하지 못한 경우, 시맨틱 매칭에서 수반하는 온톨로지의 포섭 관계(subsumption) 추론 기술을 이용하여 상위 클래스인 영화 클래스의 서비스를 발견하게 된다. 다음으로 서비스 브로커는 시맨틱 합성 기술을 이용해서 사용자의 요구를 만족시켜 주는 서비스들을 합성하게 된다. 이 과정이 서비스를 계획하는 과정이며, 본 논문에서는 계획 시스템이 이 과정을 수행한다.

### 3.2 동적 서비스 합성을 위한 구성 요소

웹 서비스 합성의 문제는 서비스의 명세의 저장소(repository)와 요청된 서비스의 질의가 주어졌는데 매칭 되는 서비스를 찾을 수 없을 때 발생한다. 그림 2는 세 개의 단일 서비스  $S_1 \sim S_4$ 로 구성된 서비스의 합성

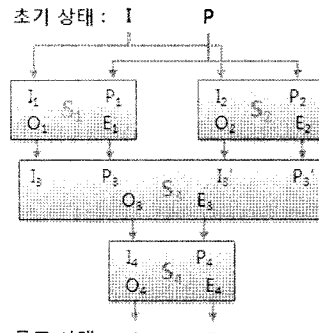


그림 2 서비스 합성 과정

과정을 보여준다.  $I$ 와  $P$ 는 현재 상황의 입력 매개변수와 전제조건이다. 그리고  $O$ 와  $E$ 는 사용자 요구의 출력 매개변수와 효과이다. 본 논문에서는 계획 시스템에 사용되는 연산자를 IOPE로 정의하고, 현재 상황을 초기 상태, 사용자 요구를 목표 상태로 정의한다. 서비스들 사이의 화살표는 입, 출력 값의 흐름이며, 예를 들어, 서비스  $S_1$ 과  $S_3$  사이의 화살표는  $S_1$ 의 출력 매개변수가  $S_3$ 의 입력 매개변수로 구성됨을 나타낸다. 다음은 웹 서비스 합성에 필요한 네 가지 정의를 설명한다. 네 번째 시맨틱 웹 서비스 합성의 정의는 웹 서비스 합성의 이해를 돕기 위해서 일반적인 탐색 방법인 순방향 탐색을 적용한 알고리즘이다.

**정의(서비스 저장소) :** 저장소는 OWL-S로 정의된 웹 서비스를 계획 시스템 연산자로 명세화한 집합을 의미한다. 하나의 서비스는 하나의 규칙(rule)으로 정의된다.

**정의(서비스) :** 유비쿼터스 환경 내에 사람의 행위를 요하는 작업(task)들을 추상화하여 '서비스'라고 정의한다. 서비스는 서비스명, 입력 매개변수, 출력 매개변수, 전제조건, 그리고 효과의 5개의 튜플(tuple)로 구성된다.  $S = (SM(I, O), [P], [E])$ 는 하나의 서비스를 표현한다. SN은 서비스 명, IOPE는 각각 서비스를 실행 할 때 필요한 입력 매개변수, 서비스를 실행한 후에 얻게 되는

출력 매개 변수, 서비스를 실행하기 위해 만족되어야 하는 전제조건, 그리고 서비스를 실행한 후에 발생하는 효과의 리스트를 의미한다.

**정의(질의):** 질의  $R$ 은 서비스 요청자의 현재 상황인 초기 상태( $P'$ )와 요청자의 요구( $E'$ )로 정의한다.  $R = ([P'], [E'])$

**정의(시맨틱 웹 서비스 합성):**  $R$ 을 만족하는 서비스들의 집합  $SS$ (Set of Services)를 구한다.

- (1) 초기화, 집합  $G = \{E_i\}$ ,  $SS = \{ \}$
- (2) 다음으로 수반하는 서비스가 있으면,  $SS$ 에  $S_i$ 을 추가한다.  
if  $(G \Rightarrow P_2)$ , then  $SS \rightarrow \{S_i\} \cup SS$
- (3) 다음으로 수반하는 서비스가 없으면, 대체 되는 서비스를 찾는다.  
else if  $(G \Rightarrow \neg P_2)$ ,  
then find  $E'_1$ ,  $G \supseteq E'_1$ ,  $G \rightarrow \{E'_1\}$   
 $\cup G \setminus \{E_i\}$ , return (2)
- (4) if  $(P' \Rightarrow P_1)$ , then  $SS \rightarrow \{S_i\} \cup SS$
- (5) if  $(P_n \Rightarrow E')$ , then  $SS \rightarrow \{S_n\} \cup SS$ , exit.

$\supseteq$ 는 연산자에 포섭하는 개체(또는 인스턴스)의 상위 개념으로의 포섭관계를 의미하고,  $\Rightarrow$ 는 수반(imply)되는 관계를 의미한다. (1)  $G$ 를 첫 번째 비교 대상이 되는 서비스의 효과로 초기화 한다. (2) 합성 단계에서 하나의 서비스는  $P_i$  뿐만 아니라 이 전 서비스로부터의

$E_{i-1}$ 를 입력 매개변수로 가진다. 예를 들어, 단일 서비스  $S_1$ 과  $S_2$ 가 합성되어야 할 서비스라면,  $S_2$ 의 효과  $E_2$ 는  $S_1$ 의 전제조건  $P_1$ 을 수반해야한다. 그림 2에서는  $S_3$ 이 해당한다. (4) 첫 번째 수행되는 서비스는 단지 질의의 초기 상태와 전제 조건이 매칭된다. 예를 들어, 그림 2의  $S_1$ 과  $S_2$ 가 해당한다. (5) 반대로 합성 과정에서 마지막으로 수행되는 서비스의 출력 값은 질의의 목표 값과 일치한다. 예를 들어, 그림 2의  $S_4$ 가 해당한다. 또한 합성 과정 중에 수행되는 특정 서비스의 효과는 다음으로 수행되는 서비스의 전제조건을 수반( $\Rightarrow$ )한다. 이 과정이 패턴 매칭이다. 반면에 이 효과가 수반하는 전제조건을 가지는 다음 서비스를 찾지 못하면 이 효과에 추론 알고리즘을 적용(단계 3)하여 이 효과에 포함되는 인스턴스와 같은 분류에 속하는 인스턴스를 추론한 후, 다음 서비스를 찾게 된다. 이 과정이 시맨틱 매칭이다.

#### 4. 시맨틱 웹 기반 상황인지 서비스를 위한 동적 서비스 제공 모델

제안하는 방법론은 서비스 공간 내에 존재하는 사용자의 상황과 서비스 시맨틱 기술을 기반으로 알맞은 서비스 혹은 세부 단위의 서비스 컴포넌트를 동적으로 합성하여 제공하는 것이 가능하다. 이를 위한 방법론은 지식 명세, 상황 정보 생성, 계획 연산자 생성, 그리고 서비스 계획 단계로 구분되며, 그림 3은 본 연구의 시맨틱 웹

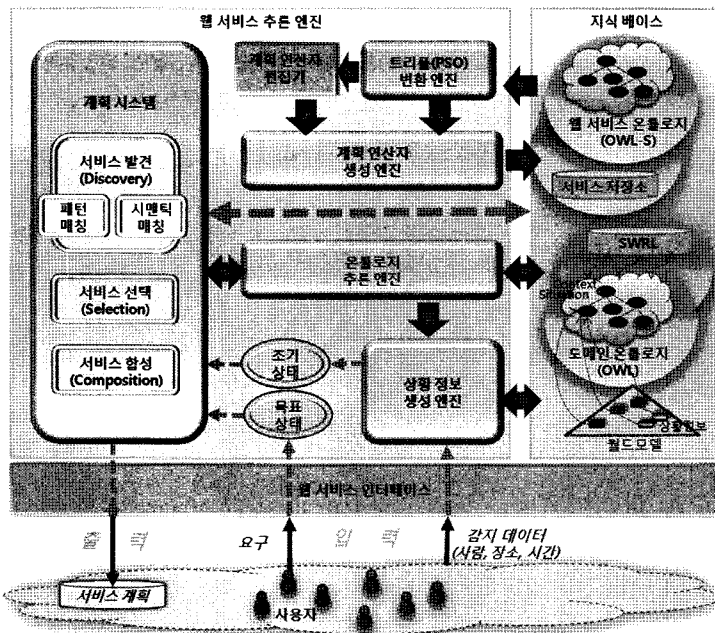


그림 3 시맨틱 웹 기반 상황인지를 위한 동적 웹 서비스 제공 모델

기본 상황인지 서비스를 위한 동적 웹 서비스 제공 모델의 구조이다. 시스템 구조를 살펴보면, 첫째, 지식 명세 단계를 통해 서비스 영역의 지식을 도메인 온톨로지로 구축하고, 그 영역에서 제공할 수 있는 서비스들을 도메인 온톨로지를 참조하여 서비스 온톨로지로 구축한다. 둘째, 상황 정보 생성 단계는 실세계에서 감지된 데이터의 월드모델에 DL 기반 추론을 적용하여 상황 정보를 생성한다. 이 상황 정보는 계획 생성 시 비교 대상이 되는 서비스의 도메인을 결정한다. 셋째, 계획 연산자 생성 단계에서는 OWL-S로 정의된 서비스의 명세를 계획 시스템에서 사용가능한 STRIPS 연산자로 변환한다. 넷째, 서비스 계획 단계에서는 현재 상황(초기 상태)에서 사용자 요구(목표)를 만족하기 위해 실행되어야 할 서비스들의 순서를 계획한다. 제안하는 계획 시스템은 단순한 패턴 매칭에 의해 서비스를 발견할 수 없을 때, 시맨틱 매칭 기술을 적용하여 대체되는 서비스를 구한다. 상황 정보 생성은 방대한 웹 서비스 명세에서 현재 상황에 가능한 서비스를 추출하는 데 효과적인 모델링 도구로 사용될 수 있다. 이 방법은 서비스 도메인으로 결정된 OWL-S 파일만을 읽어서 서비스를 추론하면, 웹 서비스 탐색 모듈에서 대부분의 시간을 차지하는 웹 서비스 로딩 시에 걸리는 시간을 많이 단축할 수 있다.

**4.1 상황정보(Context)와 상황인지(Context Awareness) 서비스**

상황인지 서비스를 위한 첫 번째 단계는 상황정보를 생성하는 것이다. 상황정보는 사용자와 컴퓨팅 환경 사이에 관련된 사용자의 환경, 객체, 상태에 관한 상황을 특징지을 수 있는 모든 정보를 말한다. 본 논문에서의 상황정보는 “웹 서비스 도메인 선택에 영향을 미치는 사용자의 특징 정보(인적 사항, 다른 사람과의 관계, 현재 장소와의 관계, 선호 정보 등)”라고 정의한다. 상황인지는 서비스 에이전트에게 실시간으로 상황 정보를 제공하는 역할을 한다. 상황인지 서비스는 서비스 에이전트가 상황을 인지하여 사용자에게 장소와 시간에 맞는 서비스를 제공하는 것을 의미한다.

**4.2 상황인지 서비스를 위한 도메인 온톨로지**

도메인 온톨로지는 상황인지 서비스 영역에서 서비스를 제공 받을 사람(사용자)과 장소, 시간에 관련된 개념(concept, 온톨로지의 클래스), 인스턴스(instance), 그리고 관계(속성)를 정의한다. 이 도메인 온톨로지는 서비스 온톨로지에 서비스 명세를 정의하기 위한 지식으로 활용되며, 핵심 온톨로지, 확장 온톨로지, 상황정보 온톨로지, 그리고 SWRL 규칙 온톨로지로 구성된다.

**핵심 온톨로지 :** 핵심 온톨로지는 서비스 영역에 서비스 요청자(requester)로 등록된 사용자가 들어오면, 센서를 통해 감지하는 정보에 관련된 지식을 정의한다. 사람(Person), 장소(Place), 그리고 시간(Time)에 대한 정

보이다. 사람 클래스는 W3C의 FOAF [12] 온톨로지를 기반으로 한다. FOAF 온톨로지는 사람에 관계되는 특징을 추출하여 65개의 클래스와 속성으로 용어를 정의하였다. 속성들은 확장 온톨로지서 도메인에 의존적인 속성들로 확장된다. 장소 클래스는 나라에 대한 정보를 포함하는 Area 클래스와 공공기관에 속하는 Institution 클래스, 그리고 FOAF 온톨로지의 Organization 클래스로 구성된다. 마지막으로 시간 클래스는 서비스의 실행 가능성을 구분하기 위해 정의한다. 예를 들어, 주치의와의 원격진료는 주치의가 병원에서 근무하는 시간 동안에만 가능하다. 이와 같은 경우 “WednesdayAfternoon”로 시간이 감지되면, 원격진료에 관련된 서비스는 실행 가능하다.

**확장 온톨로지 :** 확장 온톨로지는 핵심 온톨로지의 상위 클래스로 도메인에 의존적인 지식을 정의한다. 예를 들어, 의료 서비스를 제공한다면 핵심 온톨로지의 사람 클래스의 하위로 ‘의사’와 ‘환자’ 클래스를 정의하고, 환자와 의사와의 관계 혹은 의사의 근무지 등을 인스턴스와 속성으로 표현한다.

**상황 정보 온톨로지 :** 상황 정보 온톨로지는 상황(Situation), 상황 정보(Context), 서비스(Service) 클래스로 구성된다. 서비스 영역에서 감지된 사람(sensed-Person), 장소(sensedPlace), 그리고 시간(sensedTime) 정보가 들어오면, 이 정보를 속성 값으로 갖는 상황 클래스의 인스턴스 *I*가 생성된다. 감지된 인스턴스들은 SWRL 규칙 온톨로지에 의해 상황정보가 추가되고, 이 상황정보에 의해 *I*는 특정 서비스 클래스의 인스턴스로 추론된다. 서비스 클래스들은 상황정보 클래스의 하위에 존재하며, 감지된 정보에 대한 상황정보를 고려한 서비스 영역으로 결정된다.

- 상황(Situation) 클래스 : 환경 정보 속성으로 사람, 장소, 시간을 포함하는 감지한 상황을 묘사하는 클래스이다. 본 논문에서는 이 정보가 센서에 의해 감지되었다고 가정하고, 웹 서비스 인터페이스(그림 3)를 통해 입력한다. 이 정보가 입력되면 상황 클래스에 세 가지 속성 값을 가지는 인스턴스가 생성되고, 이는 계획 시스템의 초기 상태로 입력된다.

- 상황 정보(Context) 클래스 : 현재 상황에 실행 가능한 서비스의 영역을 결정하는 클래스이다. 추론된 서비스는 계획 시스템에서 로딩 할 OWL-S 서비스 온톨로지이며, 상황정보 온톨로지는 동적인 서비스 온톨로지의 선택을 돕는다.

**SWRL 규칙 온톨로지 :** 도메인 온톨로지에는 상황정보를 생성하기 위한 SWRL 규칙을 정의한다. 이 규칙은 핵심과 확장 온톨로지를 기반으로 감지된 환경 정보에 관련된 새로운 지식을 추론한다. 새로운 지식은 이

상황이 특정 서비스의 상황정보로 추론 되는 것을 돕는다. 이에 대한 예제는 4.4절에서 다룬다.

### 4.3 OWL-S기반 서비스 온톨로지

OWL-S는 서비스들 간의 포섭관계(subsumption) 추론이나 서비스에 명시된 속성간의 의미 관계를 추론하기 위한 명세를 정의하지 않는다. 본 연구에서는 OWL-S로 단일 서비스를 정의하며, OWL로 정의한 도메인 온톨로지의 명세를 참조하여 각 서비스의 IOPE를 정의한다. OWL 온톨로지는 지식베이스(KB)에 정의된 클래스 또는 인스턴스 사이의 새로운 관계 추론을 가능하게 하며, 이를 통해 다양한 조건의 서비스의 정의와 합성이 가능하다. IOPE 명세는 다음과 같은 규칙으로 정의된다.

#### ① 입력(I)와 출력(O) 매개 변수의 정의

입력과 출력 매개 변수는 OWL-S 온톨로지에 특정 서비스의 "hasInput"과 "hasOutput" 속성으로 각각 정의된다. 설계자가 변수명을 정의하고, 변수의 형(parameterType)을 핵심과 확장 온톨로지의 클래스를 참조하도록 정의한다. 서비스 계획 시, 참조하는 클래스의 인스턴스가 변수에 매핑 된다. 다음은 입력변수의 명세의 예를 보여준다.

```

<process:hasInput>
  <process:Input rdf:ID="Person">
    <process:parameterType rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI"
      >URI/owl_file.owl#Patient</process:parameterType>
    </process:Input>
  </process:hasInput>

```

#### ② 전제조건(P)과 효과(E)의 정의

계획 연산자 편집기는 OWL-S 파일을 읽어서 서비스들을 로딩하고, 그 서비스의 전제조건과 효과를 쉽게 편집하도록 돕는다. 전제조건은 입력변수, 효과는 출력 변수를 참조한다. 서비스 계획 시 참조하는 상수는 추론된 인스턴스도 함께 참조한다. 예를 들어, Patient ≡ hasDisease some Disease라는 Patient의 필요/충분조건에 의해 추론되는 인스턴스들도 함께 참조된다. 이 방법은 OWL-S에 추론능력을 보충하여 보다 다양하고 정확한 서비스 계획을 지원한다. 내부적으로 전제조건과 효과는 RDF 트리플(triple)로 저장되어 자동화된 서비스 계획을 지원한다. 즉, Predicate-Subject-Object(PSO) 형태로 정의되며, P는 핵심과 확장 온톨로지의 속성을 참조하고, 주체인 S와 속성의 값인 O는 입력 또는 출력 변수를 참조한다. 다음은 서비스 S<sub>1</sub>의 전제조건 중 하나의 예이다.

*person\_isAt(Patient, Departure Gps Position).*

서비스 S<sub>1</sub>이 실행되려면, 이 서비스의 다른 전제조건과 함께 환자(Patient)의 위치가 DepartureGpsPosition

에 있어야 함을 명시한다. S와 O에 해당하는 두 변수에 매핑되는 상수는 인스턴스 S의 속성 P의 값이 O이어야 한다. S와 O는 다른 서비스에 의해 동적으로 매핑되며, 위의 추론 예에서 설명한 것과 같이 추론된 인스턴스도 S에 매핑 될 수 있다. OWL-S에서 전제조건과 효과를 표현하기 위해서 권장하는 방법으로는 SWRL, KIF, DRS 등이 있다. 이 방법들은 모두 넓은 표현 범위를 갖지만, Protege와 같은 OWL-S 에디터를 통해 편집할 때 전문가의 지식을 요한다. 또한 이 표현 방법을 계획 시스템의 연산자로 변환하는 신뢰할 만한 시스템이 없다. 이에 본 연구에서는 누구나 표현하기 쉬운 인터페이스를 제공하여 서비스의 전제조건과 효과의 표현을 돕는다. 이 서비스 명세는 내부적으로 STRIPS 연산자로 정의되며, 이 과정은 STRIPS 형 계획 연산자 생성의 정확성을 높인다.

### 4.4 상황 정보 생성의 예

상황 정보는 도메인 온톨로지에 상황 인스턴스가 추가 되면 추론된다. 상황정보 추론 방법은 크게 두 단계로 나뉜다. SWRL 규칙 기반 추론 단계와 온톨로지 추론 단계이다. 첫 번째 단계는 4.2절에서 설명된 상황 클래스의 인스턴스가 생성되면 미리 정의된 SWRL 규칙을 이용하여 감지된 인스턴스들의 의미적 관계를 분석하여 새로운 지식을 추론한다. 이 지식이 상황정보이다. 두 번째 단계는 첫 번째 단계에서 추가된 새로운 지식을 포함하여 상황 인스턴스의 상황 정보에 적합한 서비스의 도메인을 추론한다. 이 때 사용되는 추론 기술은 Realization이다. 이 추론은 주어진 인스턴스가 필연적인 결과로서 속하는 클래스를 알아내는 방법이다. 즉 인스턴스의 관계(relationship)가 클래스의 필요/충분 명세(description)로부터 논리적으로 귀결(logically follow)되는지를 추론한다. DL 기반 추론엔진에 의해 가능하며, 온톨로지 지식 베이스(KB)에 정의된 클래스와 인스턴스 사이의 암시적인(implicit) 포함관계를 명시적(explicit)으로 한다. 다음은 각 단계의 예제를 설명한다.

#### SWRL 규칙의 예 :

[규칙1]

*sensedPerson(?s, ?p1) ∧ sensedPerson(?s, ?p2) ∧ differentFrom(?p1, ?p2) ∧ hasAttendingPhysician(?p1, ?p3) ∧ differentFrom(?p2, ?p3) → co-occurrenceWith(?p1, ?p2)*

[설명] 센서를 통해서 두 사람 ?p1과 ?p2가 감지된다. 도메인 온톨로지의 지식을 이용해서 ?p2가 ?p1의 주치의가 아니면, "?p1은 ?p2와 함께 있다"라는 상황정보를 추론한다.

#### Realization 추론을 위한 클래스의 예 :

Realization 추론을 하기 위해서는 클래스에 제약 조건을 명시하고, 이 조건에 맞는 인스턴스를 포함하도록 온톨로지를 설계해야 한다. 이 때, 이 클래스의 제약 조

건에 포함되는 인스턴스를 추론하기 위해서는 이 조건을 클래스의 필요/충분조건으로 명시해야 한다. 다음은 필요/충분조건을 포함하는 클래스의 예이다.

[클래스 1]

$LMCService \equiv sensedPerson \text{ some } (Patient \text{ and } (co\text{-}occurrence \text{ With some } Doctor))$   
 $sensedPlace \text{ some } Hospital$   
 $sensedTime \text{ some } OfficeHour$

[설명] 센서를 통해서 감지한 '사람1'이 환자이고, 또 다른 사람 '사람2'가 의사이면서 '사람1'과 함께 있다. 또한 감지된 장소는 병원이며, 감지된 시간이 근무시간이면 LMC(Location-based Medical Care) 서비스를 제공한다는 상황정보를 추론한다. 이 상황정보는 현재 환자가 있는 병원이 주치의가 있는 병원이 아니라는 사실을 인지(규칙1과 규칙2에 의해서)하고 원격 의료 진료 시 필요한 서비스를 추천한다.

4.5 OWL-S 기반 역방향(backward) STRIPS 계획 시스템

OWL-S 연산자는 STRIPS 시스템의 초기 모델링을 기초로 한다. 제안하는 계획 시스템은 서비스 계획의 정확성과 속도의 향상을 위해서 역방향 탐색을 이용한 STRIPS 방식을 사용한다. 역방향 탐색 방법은 주어진 목표 상태에 시작하여 초기상태에 도달하기 위해 요구되는 연산자들의 역 순서를 찾는 계획 방법이다. 이 탐색 계획 시스템은 순방향 탐색에 비해 계획 과정상의 역행(backtracking)이 발생할 가능성이 적으므로 웹 서비스 합성에 비교적 적합한 방법이다[8,13]. STRIPS [6] 방식은 일차 논리(First-Order Logic) 기반의 논리식으로 초기 상태, 목표 상태, 전제조건, 그리고 효과의 네 가지 연산자를 사용하며, 효과는 삽입(add), 삭제(delete) 리스트로 분류하여 정의한다. STRIPS는 OWL-S 프로세스 모델과 각각의 태스크들의 매칭이 가능하므로 OWL-S 기반의 웹 서비스 합성 방법으로 적합하다.

계획 연산자 생성 엔진(그림 3)은 OWL-S 명세를 STRIPS 기반 계획 시스템에서 사용하는 연산자 형식으로 변환한다. 우선, 트리플 변환 엔진을 통해 OWL-S

서비스 모델 명세를 RDF 트리플로 변환한다. 그리고 4.3절에서 논의한 계획 연산자 편집기를 통해 전제조건과 효과를 편집한다. 다음으로 계획 연산자 생성 엔진은 계획 시스템의 네 가지 연산자를 STRIPS 형태의 사실(fact) 집합으로 변환한다. 계획 시스템은 서비스 발견을 위해 패턴 매칭과 시맨틱 매칭 과정을 반복한다. 그림 4는 이 과정을 포함한 서비스합성 알고리즘을 그래프로 표현한 것이며 과정별로 살펴본다.

4.5.1 패턴 매칭

패턴 매칭은 상황을 인지하고 인지된 상황에서 사용자에게 필요한 서비스를 전제조건에 정확하게 매칭되는지를 비교하여 찾는다. 계획 시스템은 목표 상태에 만족하는 서비스(들)를 선택 할 때 까지 다음의 서비스 발견 알고리즘(그림 4)을 반복한다.

- (1) 목표(G)에 효과의 삽입 리스트(E.add)가 포함되어 있는 서비스(S)를 찾는다.
- (2) 목표에 탐색한 서비스의 전제조건(P)을 삽입하고, 목표에서 효과의 삽입 리스트를 삭제하여 목표 집합을 재생성 한다.
- (3) 목표와 초기 값이 일치 하면 종료한다.
- (4) (1)이 실패하면, 시맨틱 매칭 알고리즘을 수행한다. (1)로 돌아간다.

위의 알고리즘에서 (1)에서 (3)은 패턴 매칭 과정이다. 목표 집합을 갱신하면서 서비스가 실행되기 위해 전제조건을 만족하는 이전 서비스를 찾는 과정을 반복한다. 이 과정을 통해서 서비스들 간의 체인을 형성한다. (4)는 패턴 매칭으로 서비스(들)를 찾지 못했을 때 시맨틱 매칭 알고리즘을 호출한다. 이 알고리즘은 3.2절에서 논의한 시맨틱 웹 서비스 합성 알고리즘을 기초로 한다.

4.5.2 온톨로지 추론 기술을 이용한 시맨틱 매칭

시맨틱 매칭은 패턴 매칭에 의해서 서비스를 발견하지 못 했을 때, 온톨로지의 계층 구조를 따라가면서 동일한 효과를 낼 수 있는 대체 서비스를 찾는다. 그림 5는 4.5.1절 (4)의 시맨틱 매칭 알고리즘의 슈도코드이다. 4.5.1절의 (1)에서 (3)의 과정을 반복하면서 (1)을 만족

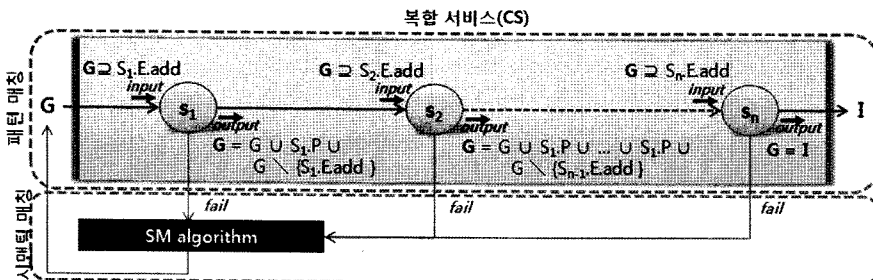


그림 4 서비스 합성 알고리즘 그래프



입력 : 패턴 매칭 되는 서비스를 탐색 할 수 없는 목표( $G$ )
출력 : 시맨틱 매칭에 의해 생성된 목표( $G'$ )
① 목표 $G = \{p_1(s_1, o_1), p_2(s_2, o_2), \dots, p_n(s_n, o_n)\}$
② while ( $p_i \neq p_n$ ) {
③     Get $p_i(s_i, o_i)$
④ $A = \{a \mid o_i \in C_j, a \in C_j, C_j \in Ot\}, (o_i \neq a)$
⑤     for ( $l = 0; l < A.SIZE; l++$ ) {
⑥ $G' \rightarrow G \cup \{p_i(s_i, A_l)\} \cup G' \setminus \{p_i(s_i, o_i)\}$
⑦         if ( $G' \supseteq S_k.E.add$ ) return $G'$
⑧         else $l = l + 1$ }
⑨ $i = i + 1$ }

그림 5 시맨틱 매칭 알고리즘

하지 않으면, 시맨틱 매칭 알고리즘을 수행하고 대안이 되는 서비스를 찾아서 (1)로 돌아간다. 시맨틱 매칭 알고리즘은 서비스들 사이의 시맨틱 관계를 유도하기 위하여 도메인 온톨로지를 통해서 매개상수를 대체할 수 있는 인스턴스를 추론 한다. 이를 위해 목표 집합  $G$ 에서 트리플을 추출하여(③) 그 트리플의 오브젝트(Object(O))와 관련된 인스턴스를 찾는다. 이 방법은 온톨로지 추론 기술 중 Realization을 이용한다. 오브젝트는 하나의 인스턴스이고, 이 인스턴스는 속성을 갖는다. 이 인스턴스의 속성과 이 인스턴스의 도메인 클래스를 필요/충분 조건으로 갖는 익명 클래스를 생성한 후, Realization 추론을 하면 오브젝트와 같은 조건을 가지는 인스턴스가 추론된다(④). 이 인스턴스를 그 트리플의 오브젝트로 대체하고(⑥) 이를 만족하는 서비스를 찾는다(⑦). 4장에서 설명한 이론은 5장의 실험 결과를 통해 타당성을 검증한다.

## 5. 실험 및 결과

제한한 온톨로지 기반 상황인지를 위한 동적 웹 서비스 제공 모델은 사용자 요구가 주어지면 현재 상황의 월드 모델로부터 요구를 만족하는 서비스들의 순서를 계획한다. OWL-S로 정의된 서비스들을 계획 시스템에서 사용하는 연산자로 자동 변환하고 이를 계획 시스템이 사용한다. 제한한 방법의 우수성 검증 실험을 위해 일반적인 웹 서비스 모델을 보여준 기존연구 OWLS-XPlan[1]과 본 연구의 동적 모델링 방법을 동일한 시나리오와 서비스를 통해 어떻게 웹 서비스의 합성과 실행 시 상황 정보를 고려하여 맞춤형 서비스를 제공하는지 예제를 통해 보이고, 비교 분석하여 본 논문의 우수성을 확인한다.

### 5.1 시나리오

본 연구는 독일의 BMB+F 프로젝트의 실험 시나리오로 OWLS-XPlan 시스템에서 예제로 웹서비스의 적용 가능성을 입증한 '위급 환자 이송' 시나리오<sup>2)</sup>를 응용하여 상황에 필요한 서비스들을 계획하는 예를 보인다. 웹 서비스 시스템에 등록된 사용자가 여행 중 상해를 입어 병원을 찾는다. 병원1의 의사는 환자(사용자)가 지병이 있음을 알고 주치의가 있는 병원2로 환자를 이송시키는 웹 서비스를 선택한다. 웹 서비스 시스템은 현재 상황과 환자가 필요한 요구사항을 가지고 환자에게 제공할 서비스를 계획한다.

### 5.2 웹 서비스 인터페이스

본 연구의 시스템 구조(그림 3)의 웹 서비스 인터페이스는 3.1절에서 살펴본 서비스 브로커의 역할을 한다. 웹 서비스 인터페이스는 사용자 요구에 적합한 서비스를 제공하기 위해 사용자의 현재 상황을 인지하고 상황에 맞는 서비스를 제공하기 위한 서비스들의 경로 계획을 제공하는 인터페이스이다.

그림 6은 상황을 인지하여 서비스 온톨로지를 시스템에 적재하고, 서비스 온톨로지를 기반으로 서비스 연산자를 편집할 수 있는 인터페이스를 보여준다. 제한한 모델링 시스템은 온톨로지 파서(parser), 온톨로지 추론 엔진, 그리고 서비스 계획 모듈들이 기반으로 하는 로직의 구문(syntax)과 의미(semantics)가 깨지지 않게 하기 위해 각 모듈이 기반으로 하는 자바(Java) 프로그래밍 언어를 사용하여 구축하였다. 서비스 인터페이스는 세 가지 컴포넌트로 구성된다. 현재 상황(Situation)의 환경 정보를 획득하는 컴포넌트, 환경 정보에 대한 상황 정보(Context)를 추론하는 컴포넌트, 그리고 서비스(Service)를 계획하는 컴포넌트이다. 4장에서 설명한 이론을 기반으로 각 컴포넌트의 기능과 시뮬레이션 예를 설명한다.

#### 5.2.1 상황인지를 위한 상황 정보 추론

첫 번째 컴포넌트의 입력은 실제 환경에서 감지한 정보이다. 현재 시스템은 웹 서비스 계획 모델링에 초점을 두어 다양하고 정확한 값으로 시뮬레이션이 쉽게끔 감지 정보를 수동으로 입력하도록 하였다. 이 정보는 도메인 온톨로지에 정의된 지식을 기반으로 현재 상황에서 추출할 수 있는 상황정보를 추론하여, 상황정보 컴포넌트에 추론된 서비스 영역을 출력하고, 그 서비스 영역에 해당하는 OWL-S 파일(들)을 시스템에 적재한다.

그림 6의 상황 컴포넌트에서 선택하는 인스턴스들은 상단의 "File" 메뉴를 통해 도메인 온톨로지를 읽어서, 사람, 장소, 시간 클래스의 인스턴스 목록을 보인 것이다. 이를 위해 추론 엔진과 연동이 쉬운 Protege-OWL

2) [http://www-ags.dfki.uni-sb.de/~klusuch/scallops/html/scenario\\_1.html](http://www-ags.dfki.uni-sb.de/~klusuch/scallops/html/scenario_1.html)

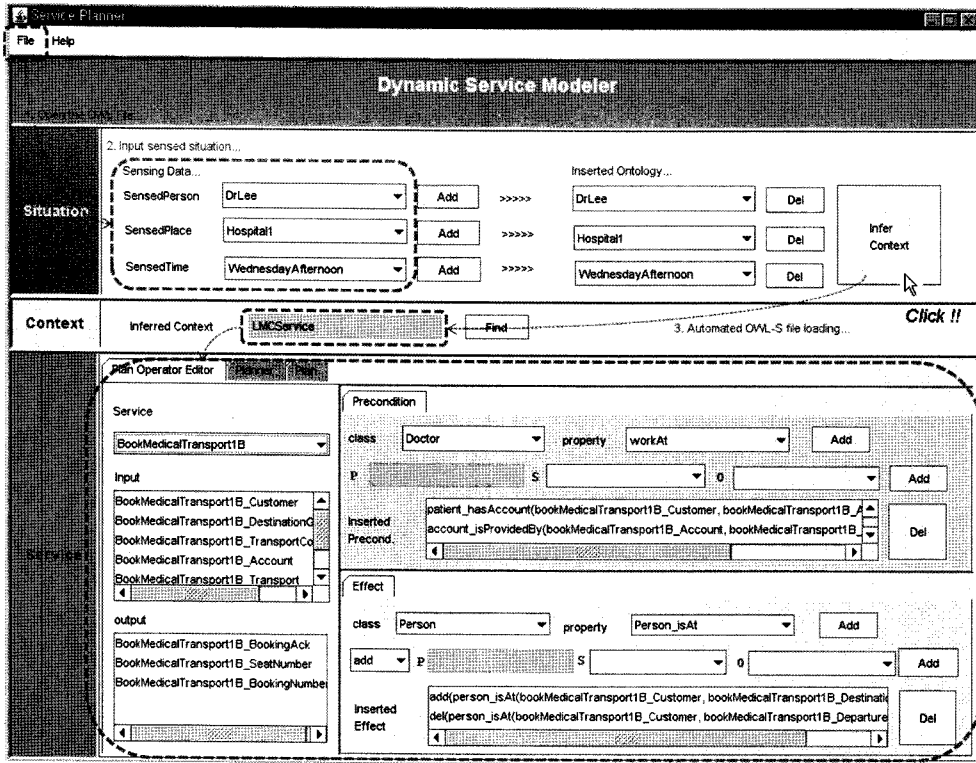


그림 6 동적 웹 서비스 모델링 시스템

API<sup>3)</sup>를 사용하였다. 그림 6의 시뮬레이션을 기반으로 실제 추론 과정을 설명하면, 감지된 환경 정보로 사람은 *Mikka*와 *Dr.Lee*, 장소는 *Hospital1*, 그리고 시간은 *WednesdayAfternoon*이다. 이 환경 정보는 도메인 온톨로지의 지식 베이스를 참조하여 두 번째 컴포넌트에 출력될 서비스 영역을 추론한다. 이 과정은 4.2절에서 설명한 도메인 온톨로지를 기반으로 다음과 같은 순서로 실행된다.

- (1) 상황 클래스에 감지된 환경 정보를 가지는 인스턴스를 생성한다(표 1의 ①~⑤).
- (2) 감지된 사람 중 *Mikka*는 웹 서비스 시스템에 등록된 사용자이다. 표 1의 ⑥, ⑦은 *Mikka*와 관련된 명시된 지식이다. 표 1과 2는 지식을 DL 기반 Predicate 형식으로 나타낸다. 표 1은 도메인 온톨로지의 특정 클래스에 명시한 인스턴스 지식이고, 표 2는 온톨로지 추론을 통해서 추론된 지식이다.
- (3) SWRL 규칙 온톨로지를 기반으로 새로운 지식을 추론한다. 4.4절에서 설명한 두 개의 규칙 중, *Dr.Lee*가 *Mikka*의 주치의가 아니므로 [규칙1]을 만족하여 “*Mikka*는 *Dr.Lee*와 함께 있다”라는 상

황 정보를 추론한다. 이에 의해 표 2의 지식①이 KB에 추가된다.

- (4) 필요/충분조건을 포함한 클래스를 기반으로 새로운 지식을 추론한다. 4.4절에서 논의한 두 개의 클래스 중 [클래스1] 조건을 만족한다. 즉, 환자 *Mikka*는 의사와 함께 있고(표 2의 ①), 장소는 병원이며, 근무시간이므로 LMC 서비스를 추론한다(표 2의 ②).

(1)에서 (4)까지의 과정을 통해 두 번째 상황 정보 컴포넌트의 LMC OWL-S 서비스가 시스템에 적재된다. 이 과정의 추론은 OWL-DL 추론 엔진으로 유명한 Pellet의 프로그램 코드를 사용한다. Pellet은 SHOIQ DL의 추론을 위해 개발된 tableaux 알고리즘에 기초하며 taxonomy 분류의 확인을 위한 complete 문법을 가진다. 또한 SWRL 추론을 지원하여 본 논문의 추론 모듈로 적용하기에 적합하다[14].

#### 5.2.2 서비스 명세 편집 및 계획

웹 서비스 인터페이스의 세 번째, 서비스 컴포넌트는 두 개의 모듈로 구분된다. 계획 연산자 편집기(Plan Operator Editor 탭)와 계획 시스템(Planner 탭)이다. 계획 연산자 편집기는 그림 6의 서비스 컴포넌트에 보여 지는 부분이다. 계획 연산자 편집 시에는 상황 정보

3) <http://protege.stanford.edu/plugins/owl/api/>

표 1 도메인 온톨로지에 명시된(explicit) 지식

① <i>Situation</i> ( <i>Situation 13</i> ) ② <i>sensedPerson</i> ( <i>Situation13, Mikka</i> ) ③ <i>sensedPerson</i> ( <i>Situation13, Dr Lee</i> ) ④ <i>sensedPlace</i> ( <i>Situation13, Hospital 1</i> ) ⑤ <i>sensedTime</i> ( <i>Situation13, WednesdayAfternoon</i> )	감지된 환경 정보를 가지는 인스턴스 <i>Situation13</i> 의 지식
⑥ <i>Patient</i> ( <i>Mikka</i> ) ⑦ <i>hasAttendingPhysician</i> ( <i>Mikka, Dr Kim</i> )	<i>Mikka</i> 는 환자이다. <i>Mikka</i> 의 주치의는 <i>Dr Kim</i> 이다.
⑧ <i>Doctor</i> ( <i>Dr Kim</i> ) ⑨ <i>Doctor</i> ( <i>Dr Lee</i> ) ⑩ <i>Hospital</i> ( <i>Hospital 1</i> ) ⑪ <i>Office Hour</i> ( <i>WednesdayAfternoon</i> )	<i>Dr Kim</i> 은 의사이다. <i>Dr Lee</i> 는 의사이다. <i>Hospital 1</i> 은 병원이다. <i>WednesdayAfternoon</i> 은 근무시간이다.

표 2 도메인 온톨로지에 추론된 지식

① <i>co-occurrence With</i> ( <i>Mikka, Dr Lee</i> )	<i>Mikka</i> 는 <i>Dr Lee</i> 와 함께 있다.
② <i>LMCService</i> ( <i>Situation13</i> )	

컴포넌트의 "Find" 버튼을 눌러 OWL-S 서비스 파일을 수동으로 적재할 수 있다. 5.3절 그림 7에서 예제를 통해 이 경우를 보여준다. 그림 6에서는 상황인지 후, OWL-S 파일을 자동으로 적재하여, 그 파일에 해당하는 서비스의 목록을 왼쪽 상자에 보여주고, 오른쪽 상자에는 그 서비스의 전제조건(Precondition)과 효과(Effect)를 보여준다. 이 전제조건과 효과는 계획 시스템에 연동되는 서비스 저장소로부터 읽어 온 것이다. 사용자는 이전에 정의한 서비스 명세를 이 브라우저를 통해 확인하고 편집할 수 있다. 왼쪽 상자에서, 서비스명에 따른 입출력 매개 변수들은 서비스의 전제조건과 효과를 편집하는데 사용된다. 입력 변수는 전제조건 편집 시에 사용되며, 출력 변수는 효과 편집에 사용된다. OWL-S 서비스 온톨로지의 파싱은 Pellet 추론엔진과의 통일성을 위해 Jena[15]를 사용하였다.

계획 시스템은 상황 컴포넌트의 입력 값으로 인해 OWL-S 서비스 온톨로지를 자동으로 추론하여, 그림 6의 상황 컴포넌트의 LMC 서비스와 같이 출력하고 적재한다. 상황 정보와 관련된 하나 이상의 서비스 온톨로지가 적재되며, 상황 정보에 따른 서비스 온톨로지 목록은 텍스트 파일로 따로 관리된다. 계획 시스템은 초기 상태(현재 상황)와 목표 상태(사용자 요구)를 입력 받아서 초기 상태에서 목표 상태를 만족하는 서비스들을 계획한다. 계획 시스템은 계획 연산자 편집기와는 달리 감지된 정보를 바탕으로 상황 정보(Inferred Context; OWL-S 파일)가 동적으로 추론되고, 서비스 온톨로지가 적재되어야 작동이 가능하다. 이 편집기 역시 서비스 컴포넌트의 하나의 탭으로 위치한다(그림 6의 Planner 탭). 초기 상태와 목표 상태의 입력 형식은 동일하다. 초기 상태는 목표에 대응되는 현재의 상황을 입력한다. 이는 상황 정보 추론으로부터 추가된 지식을 이용하여 정의 한다. 입력된 초기와 목표 상태는 계획 시스템 지식 베이스에 저

장되고 그림 9의 Plan 탭에 출력된다.

제한한 계획시스템은 SWI Prolog<sup>4)</sup>를 사용하여 웹 서비스 계획 시스템의 프로토타입을 구현하였다. Prolog는 DL-Predicate 기반 패턴 매칭에 적합하며, 더욱이 STRIPS 기반 역방향 탐색을 위해 지식들의 의미 표현을 유지하면서 빠른 속도와 정확성을 지원한다. 계획 시스템의 초기와 목표 상태는 사용자에게 의해 동적인 입력이 가능하며, 제안한 인터페이스는 온톨로지의 지식을 사용하여 STRIPS 기반 계획 시스템의 입력 연산자를 동적으로 생성하기에 적합하다.

5.3 웹 서비스 명세

서비스 명세는 OWL-S 온톨로지로 정의한다(4.3절 참조). OWL-S 프로세스 모델에는 서비스명, 입력과 출력 매개변수가 정의된다. 5.1절의 시나리오를 위한 서비스들 중 서비스 "BookMedicalTransportIB"를 예제로 서비스 명세를 정의하는 방법을 설명한다. 서비스 모델은 속성 "describedBy"로 서비스와 연결되어 서비스 작동 명세를 IOPE로 표현한다. 다음으로 계획 연산자 편집기(그림 7)는 서비스 모델의 명세를 읽어서 서비스명과 입출력 변수를 출력한다(그림 7의 왼쪽 탭). 사용자는 서비스 명세를 참조하여 그 서비스의 전제조건과 효과를 편집 할 수 있다(그림 7의 오른쪽 탭). 이 IOPE 명세는 서비스를 계획하는 데 연산자로 사용되며, 계획 시스템의 지식베이스에 저장된다. 계획 연산자 편집기를 통해 OWL 온톨로지, OWL-S 온톨로지, 그리고 계획 시스템은 서로의 지식 베이스를 공유한다. 이 시스템을 이용하면, 하나의 인터페이스에서 서비스 편집과 계획이 가능하다. 전제조건과 효과의 편집 방법은 동일하나, 효과에 정의되는 명세는 삽입(add) 인지 삭제(del) 리스트 인지를 명시 하여야 한다. 초기와 목표 상태의 입력 형

4) <http://www.swi-prolog.org/>

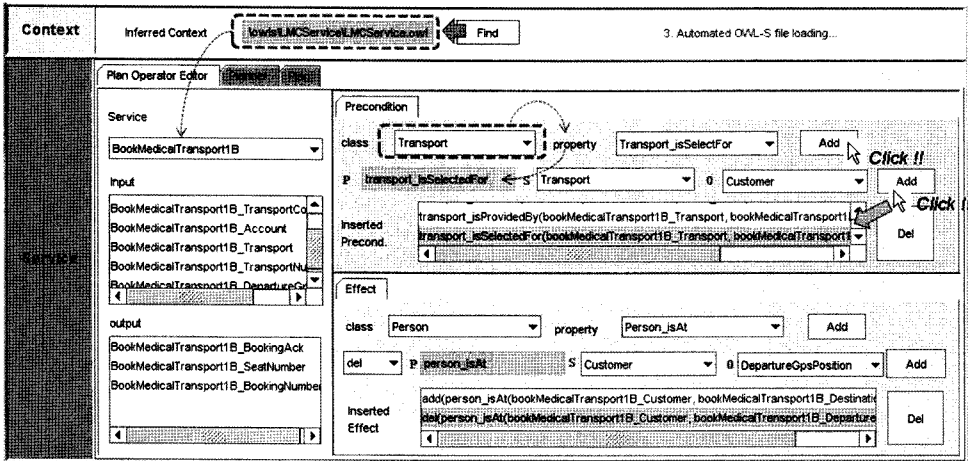


그림 7 전제조건과 효과 편집의 예

식과 비슷하며, 클래스를 선택하고 속성을 선택하면, 이 속성이 P로 설정되어, S와 O를 입력할 수 있는 인터페이스를 제공한다. S와 O는 인터페이스의 디자인을 고려하여 서비스 명(BookMedicalTransport1B)을 제외한 “\_” 뒤의 변수명만 보이게 하였다. 이와 같은 과정으로 편집된 서비스는 계획 시스템의 서비스 저장소에 3.2절에서 설명한 서비스 명세대로 저장된다. 저장소에는 그림8의 “Inserted Precond.”에 보이는 것과 같이 서비스 명을 생략하지 않은 변수명이 삽입된다. 다음은 서비스 저장소에 저장된 “BookMedicalTransport1B” 서비스 명세의 일부를 보여준다. 이 역시 지면 관계상 서비스명은 생략하였다.

```
rule(bookMedicalTransport1B()),
[account_isProvidedBy(_Account, _Transport Company),
transport_isSelectedFor(_Transport, _Customer), ...]
[add(person_isAt(_Customer, _Destination GpsPosition)),
del(person_isAt(_Customer, _Departure GpsPosition))].
```

5.4 온톨로지 기반 시맨틱 매칭을 이용한 서비스 계획

본 논문에서 제안하는 온톨로지 기반 시맨틱 매칭 방법의 성능을 객관적으로 비교하기 위해 기존 연구인 OWLS-XPlan에서 실험 예제로 사용한 온톨로지 확장하여 성능 평가 기준으로 사용하였다. OWLS-XPlan은 본 논문의 2.2절에서 설명한 바와 같이 서비스 계획 시에 전달되는 서비스들의 IOPE를 고려하여 온톨로지와 서비스를 모델링 하였다. 이 방법은 서비스 실행 시 적용되는 입력 값을 온톨로지 클래스의 인스턴스로 미리 정의 해놓고, 특정 서비스의 전제조건으로 대입되어 그 서비스가 실행 되도록 한다. 이 정적인 모델링은 각 서비스의 전제조건에 포함되는 속성들을 온톨로지에 미리 정의해야 하는 작업이 요구된다. 또한 단순 패턴 매칭으

로 서비스를 탐색하므로 IOPE를 만족하는 서비스가 존재하지 않으면 대체 서비스를 찾을 수가 없다. 본 비교 실험에서는 객관적인 평가를 위해 OWLS-XPlan의 계획 경로를 본 논문에서 제안한 방법으로 계획한다. 따라서 본 논문의 성능 평가를 위한 실험 결과는 서비스 온톨로지, 그리고 초기와 목표 상태를 동일하게 하고, 도메인 온톨로지는 상황 인지와 시맨틱 기술을 위해 확장하여 실험 하였다.

그림 8은 실험에서 사용한 시맨틱 매칭이 필요한 시나리오의 예이다. 전제조건과 효과는 DL-Predicate 형식으로 변수는 대문자로, 상수는 소문자로 명칭한다. 3.2절에서 설명한 바와 같이 초기 상태에서 목표 상태를 만족하기 위한 하나의 서비스를 찾지 못 할 경우에는 서비스들을 합성하여야 한다. 이에 대상이 되는 서비스의 전제조건을 만족하기 위해서 다른 선택되는 서비스가 필요하다. 그림 8의 ⑧에서 ⑫의 서비스들은 초기 상태에서 목표 상태를 만족하기 위해 사용자의 위치를 변경하는 효과를 포함 있다. 하지만 이들 서비스의 전제조건을 만족하기 위해서는 ①에서 ⑦의 서비스가 선행 되어야 한다. 예를 들어, 서비스 ⑧은 사용자의 위치(isAt)를 hospital1에서 faro로 옮긴다. 하지만 이 서비스의 전제조건에는 사용자의 현재 위치(isAt)를 출발지(DepartureGpsPosition)로 갖는 교통편을 선택하는 서비스가 선행 되어야 한다. 이에 ①의 서비스를 실행한 후 나온 효과로 서비스 ⑧의 전제조건을 만족하여 실행하게 된다. 하지만 ⑧에서 ⑫의 서비스는 ⑪과 ⑫의 서비스가 연결되지 않아서, 초기상태로부터 목표 상태를 만족하는 계획을 생성할 수 없다. OWL-XPlan은 정적인 서비스 모델에 단순한 패턴 매칭 방법을 사용하여 이 시나리오를 실행하면 에러가 발생한다. 본 논문에서

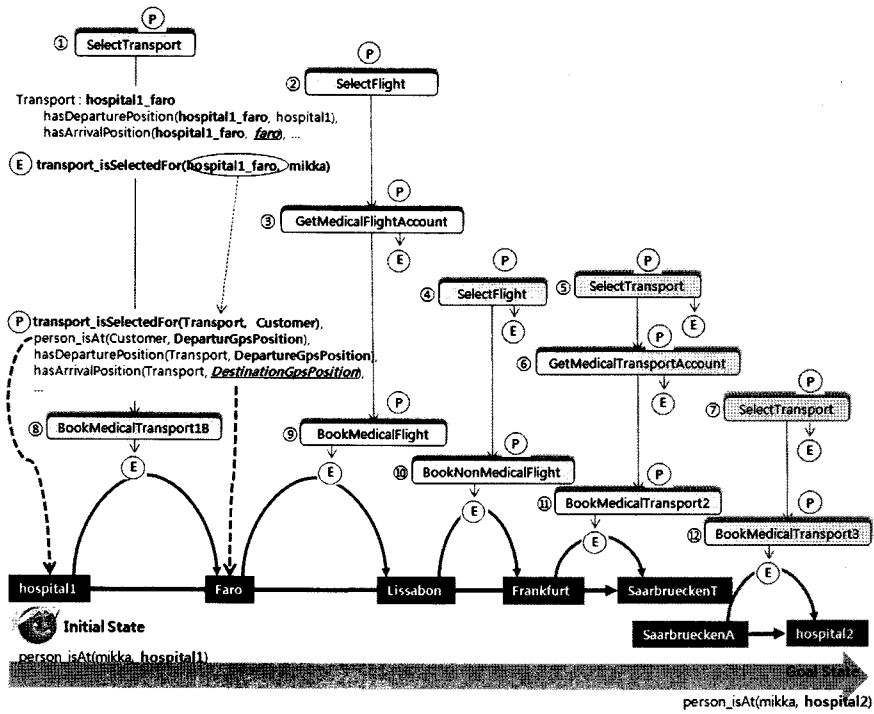


그림 8 시맨틱 매칭이 필요한 서비스 명세의 예

는 4.5절에서 설명한 OWL-S 기반 역방향 STRIPS 계획 시스템을 사용하여 이 문제를 해결하였다.

5.5 실험 결과

그림 9는 그림 8의 서비스 명세에 따른 시맨틱 매칭을 이용한 서비스 합성의 결과이다. 각 서비스의 효과의 add 리스트를 서비스명 하단에 출력하도록 하였다. 그림 9에서 볼 수 있듯이 mikka의 초기 상태 위치인 hospital1에서 목적지인 hospital2를 만족하는 상태까지 서비스 계획이 생성된 것을 볼 수 있다. 서비스 (1)에서 (12)까지를 계획하는 알고리즘은 4.5절에서 제안한 역방향 탐색 알고리즘을 기반의 시맨틱 매칭 방법을 사용한다. 목표 상태를 만족하는 서비스를 찾고, 역으로 초기 상태를 만족할 때까지의 서비스를 구한다. OWLS-XPlan과 비교하여 제안하는 매칭 방법은 그림 8과 같이 매칭 되는 서비스가 없게 모델링된 예의 서비스 탐색을 가능하게 한다.

시맨틱 매칭이 필요한 부분을 살펴보면, 역방향 탐색 기법을 사용하여 첫 번째로 목표 상태를 만족하는 서비스 (7)과 (12)를 찾는다. 서비스 (7)은 SaarbrueckenA에서 hospital2로 가는 교통편을 선택하는 서비스이고, (12)는 mikka를 SaarbrueckenA\_hospital2 교통편을 이용하여 이동시키는 서비스이다. 다음으로 SaarbrueckenA를 목적지로 갖는 교통편을 찾는다. 하지만 이러한 효과를 수

반하는 서비스가 정의되어 있지 않다. 제안하는 방법은 패턴 매칭 되는 서비스가 없을 때, 대안이 되는 서비스를 찾는다. 서비스 (7)과 (12)를 선택하여 목표 상태를 만족하고, 시맨틱 매칭을 통해 현재 상태를 만족할 서비스로 (5)와 (11)을 선택한다. 교통편의 경로를 살펴보면, (5)에 의해 Frankfurt→SaarbrueckenI, (7)에 의해 SaarbrueckenA→hospital2가 된다. 그리고 실제 위치 이동은, (11)에 의해 Frankfurt→SaarbrueckenI, (12)에 의해 SaarbrueckenA→hospital2가 되어 초기 상태에서 목표 상태를 만족하는 경로를 완성한다.

본 연구에서는 시맨틱 서비스 모델링을 위해서 OWLS-XPlan의 초기 온톨로지와 목표 온톨로지를 하나로 통합하여 도메인 온톨로지로 정의하고, 시맨틱 추론을 위한 클래스 구조와 인스턴스간의 의미 관계를 확장한다. 도메인 온톨로지는 도메인의 존재하는 지식들의 포섭관계와 지식 간의 관계들을 명시하여 서비스 계획에서 필요한 서비스를 못 찾게 되면 도메인 온톨로지 구조를 참조하여 대안이 되는 서비스를 찾는다.

도메인 온톨로지의 일부를 표현한 그림 10을 통해 이 원리를 살펴보면, 서비스 ⑫는 목표 상태인 person\_isAt(mikka, hospital2)를 효과로 가지고, 전제조건으로 mikka의 현재 위치가 SaarbrueckenA 일 때 만족한다. 이 서비스는 목표 상태로 만족하는 서비스이므로 실행

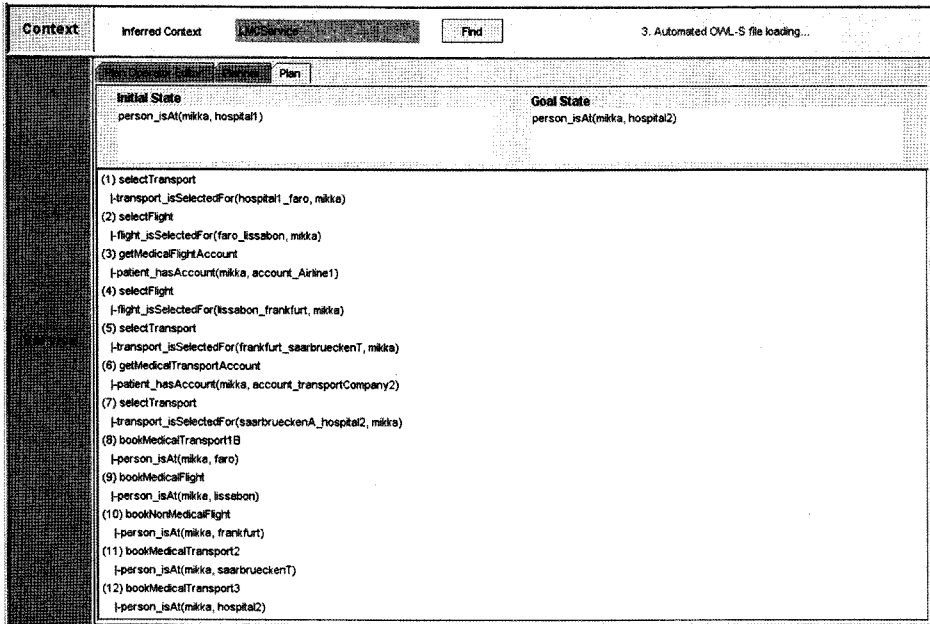


그림 9 계획 시스템의 서비스 계획 결과의 예

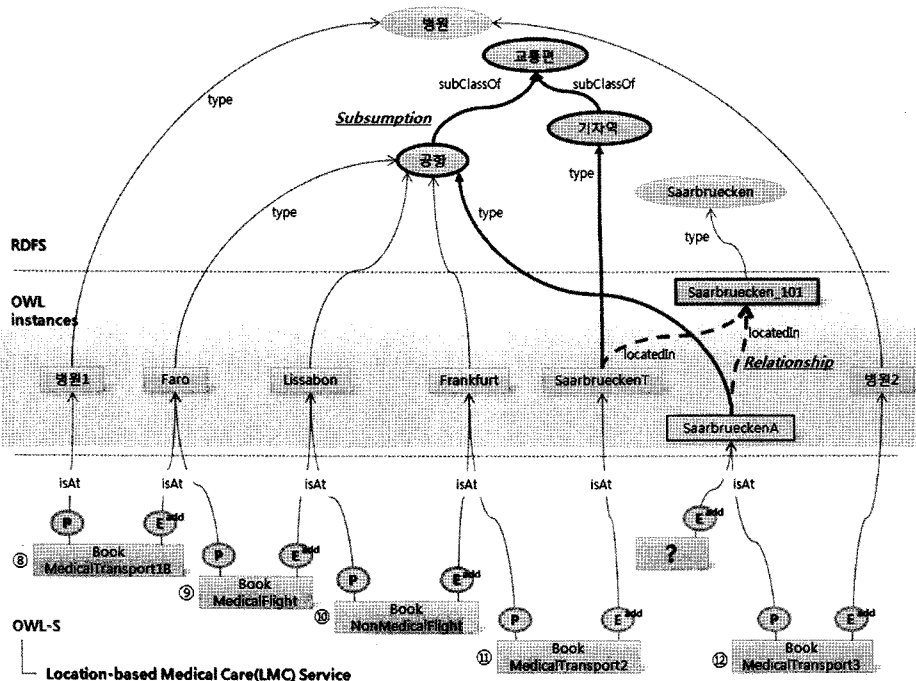


그림 10 온톨로지 기반 시맨틱 매칭의 예

가능한 서비스로 선택된다. 다음으로 서비스 ⑫의 전제 조건을 만족하는 효과를 갖는 서비스를 찾아야 하는데 가능한 서비스가 없으므로 시맨틱 매칭을 한다. 시맨틱 매칭은 4.5.2절에서 설명한 바와 같이, 익명의 클래스를

만들고 현재 문제가 되는 인스턴스의 특징과 유사한 인스턴스를 추론한다. 이때 사용하는 온톨로지 Realization 기술은 sound하고 complete 하다[16]. 시맨틱 매칭 기술은 모든 암시적인(implicit) 포섭관계 또는 인스턴스

관계를 결정 가능하며, 이는 동적 서비스 모델링의 큰 장점이다.

서비스 ⑫의 전제조건을 보면, 사용자의 현재 위치(isAt)가 Saarbruecken.A이다. 이를 만족하는 효과의 서비스가 없으므로, 다음과 같은 익명의 클래스를 자동 생성하고, 이 조건에 속하는 인스턴스들을 Realization 추론한다.  $Anon\_19 \equiv Vehicle \sqcap (locatedIn \text{ has } Saarbruecken\_101)$

Anon\_19는 Saarbruecken.A의 특징을 기술한다. 그림 10의 온톨로지 구조에서 보듯이, Saarbruecken.A는 공항이면서 Saarbruecken.A\_101에 위치(locatedIn)한다. 온톨로지 구조를 따라가 보면, 공항은 교통편(Vehicle)에 속한다(Subsumption). 이 조건을 갖는 Anon\_19 클래스를 생성하고 이 클래스에 속하는 인스턴스를 추론하면 Saarbruecken.T라는 인스턴스를 추론 할 수 있다. 이 인스턴스를 효과로 포함하는 서비스를 검색하면, 서비스 ⑪을 추출할 수 있으며 이로 인해 초기 상태에서 목표 상태를 만족하는 서비스들의 경로를 계획할 수 있다.

**5.6 비교 실험 결과**

이 절에서는 도메인이 다른 예제를 통해 제안한 동적인 서비스 모델링의 우수성을 입증한다. 이 시나리오는 3.1절에서 살펴본 “뮤직 비디오 재생”이라는 사용자 요구를 만족하기 위한 서비스들을 계획하는 예제이다. 5.1절 시나리오의 시뮬레이션과 마찬가지로 동적 웹 서비스 모델링 시스템(그림 6)에 감지된 정보를 입력한다. 감지된 정보는 다음과 같다.

- ① *Situation(Situation55)*
- ② *sensedPerson(Situation55, Michael)*
- ③ *sensedPlace(Situation55, Room003)*
- ④ *sensedTime(Situation55, SaturdayNight)*

제안한 시스템은 도메인 온톨로지 지식을 기반으로 감지된 정보를 통해 상황정보를 추론한다. 도메인 온톨

로지에 Room003의 주인(owner)이 Michael이라는 사실을 통해 다음 규칙이 실행되어 occupiedMy(Michael, Room03)이라는 상황정보를 추론한다.

$$sensedPerson(?s, ?p) \wedge sensedPlace(?s, ?l) \wedge owner(?l, ?p) \rightarrow occupiedMy(?p, ?l)$$

이 상황정보는 다음 규칙에 의해 서비스 Home-Service가 추론되어, 이 서비스 영역에 해당하는 OWL-S 파일이 시스템에 적재된다.

$$HomeService \equiv sensedPerson \text{ some } (Person \text{ and } (occupiedMy \text{ some } Place)) \text{ sensedPlace \text{ some } Home}$$

다음으로 사용자는 초기상태와 목표상태를 입력하여 관련된 서비스를 요청한다. 이 예제에서는 그림 9와는 다르게 초기상태와 목표상태의 속성이 다르더라도 계획이 가능함을 보여준다. 비교연구인 OWLS-XPlan 1.0은 속성이 동일할 경우만 가능하며, 2.0 버전은 2.2절에서 언급한대로 부과적인 절차가 요구되어 동적인 처리가 쉽지 않다.

그림 11은 Michael이 room003에 있는 초기 상태로부터 Michael이 요구하는 뮤직 비디오 m\_t\_crazy001을 재생하기 위한 서비스 계획 결과를 보여준다. 실험 결과를 보면, 뮤직 비디오 서비스가 아닌 영화 서비스가 실행된 것을 볼 수 있다. HomeService 서비스 온톨로지에는 뮤직 비디오를 실행하는 서비스를 정의하지 않았다. 하지만 제안한 시스템은 시맨틱 매칭을 통해 대체 서비스를 탐색하고 실행하여 가장 비슷한 서비스 결과를 제안한 것을 볼 수 있다. 원리를 살펴보면, 서비스 ‘movie\_play’의 효과인 watch(Person, Movie\_title)의 object, Movie\_title는 도메인이 영화(Movie)로 정의되어 패턴매칭으로는 탐색되지 않는다. 하지만 그림 5의 알고리즘에 의해 시맨틱 매칭이 실행되어, 뮤직 비디오의 상위 개념인 영화의 서비스를 검색하고, 뮤직 비디오 인스턴스 ‘m\_t\_crazy001’이 매핑 되어, 초기 상태로부터

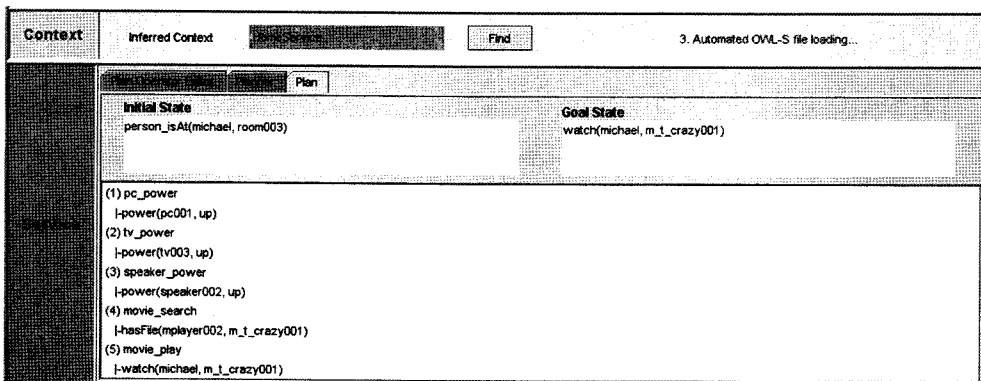


그림 11 계획 시스템의 서비스 계획 결과 예 2

결과를 유도하는 경로를 구하게 된다. 서비스 (1)에서 (4)는 서비스(5)의 전제조건을 만족하기 위해 선행되어야 하는 서비스들의 실행 순서를 보여준다. 도메인 온톨로지를 통해서, 감지된 사용자 위치에 존재하는 관련 디바이스와 그에 대한 서비스들을 제시한다. OWLS-XPlan에서는 서비스 설계자가 서비스의 합성을 고려하여 정의한 서비스의 최초 명세에만 국한된 획일화된 서비스 결과를 제공하는 것에 비하여, 본 연구에서는 온톨로지 기반의 시맨틱 매칭을 이용하여 동적인 서비스 합성을 제공한다. 제안한 모델은 동적인 웹 서비스의 제공을 위해 사용자가 존재하는 서비스 영역의 상황을 인지하여 검색해야 하는 서비스의 목록을 자동으로 추론하였고, 이는 대용량의 서비스 명세를 분석하는 시간을 단축하였다. 또한 계획 시스템의 실행 시간에 초기 상태와 목표 상태를 입력 할 수 있게 하여, 동적인 상황에 대한 실행 가능한 서비스의 계획을 제시하였다. 그리고 특정 조건과 효과를 포함한 서비스가 정의되어 있지 않을 때, 시맨틱 매칭을 통해 대안이 되는 서비스를 추론하게 하여 완전하지 않은 서비스 저장소의 단점을 해결 하였다.

## 6. 결론 및 향후 연구

본 논문은 사용자 요구와 감지한 상황의 월드 모델을 분석하여 계획 시스템에 목표와 초기 상태로 입력하고 초기 상태로부터 목표를 만족하기 위한 일련의 작업들을 계획하는 동적 서비스 제공을 위한 모델링 방법을 제안한다. 제안한 방법은 시맨틱 웹 기반의 상황인지 기술을 사용하여 방대한 서비스 저장소에서 상황에 필요한 서비스 명세만 탐색한다. 또한 시맨틱 매칭 기술에 의해 요구되는 서비스 명세가 존재하지 않더라도 대안이 되는 서비스를 발견한다. 이렇게 제안된 시스템은 OWLS-XPlan에서 지원하지 못하는 시맨틱 매칭과 동적인 초기 상태와 목표 상태의 입력에 따른 서비스 계획을 가능하게 하였다.

기존의 시맨틱 웹 기반의 서비스 합성 방법들은 서비스 영역 지식의 설계, 서비스의 명세, 그리고 이러한 지식이 존재하는 영역에 사용자가 접근하였을 때, 이 지식의 표현력을 유지하면서 계획 시스템과 결합하여 적합한 서비스를 추론하는 동적인 서비스 모델 구조를 다루지 못했다. 하지만 본 논문은 시맨틱 웹 기반 기술인 OWL 온톨로지, OWL-S 온톨로지에 정의된 지식들의 표현력을 유지하면서 계획 시스템의 STRIPS 연산자로 변환하여 지식 베이스를 공유하는 모델을 설계하였다. 제안한 방법은 현재 상황(초기 상태)이 인지되고, 사용자 요구(목표 상태)만 들어오면, 그 상황에서 가장 적당한 하나의 서비스 혹은 서비스들의 실행 순서를 계획할 수 있다.

향후 연구로는 본 연구에서 제안한 방법론을 복합 서비스 단위로 웹 서비스 명세를 확장하여 좀 더 실현 가능한 시스템을 구축하는 것이다. 또한 전제조건과 효과의 표현으로 OWL-S 1.1에서 권장하는 방법인 SWRL 또는 KIF 등을 이용한 동적인 모델링을 제안하는 것이다. 아직 이 기술은 표준화와 계획 연산자로 변환하는데 문제가 많지만 시스템과 유기적인 연동을 위해서는 반드시 필요한 연구이다.

## 참고 문헌

- [1] Klusch, M., Gerber, A. and Schmidt, M., "Semantic Web Service Composition Planning with OWLS-XPlan," *Proc. of the AAAI Fall Symposium on Semantic Web and Agents*, Arlington VA, USA, AAAI Press, 2005.
- [2] Nau, D., Au, T., Ilghami, O., Kuter, U., Murdock, J., Wu, D. and Yaman, F., "SHOP2: An HTN planning system," *Journal of Artificial Intelligence Research* 20 (2003) 379-404. 2003.
- [3] Naveen, S., Massimo, P. and Katia, S., "Semantic Web Service Discovery in the OWL-S IDE," *Proc. of the International Conference on System Sciences*, 2006.
- [4] David, M., Mark, B., Drew, M., Sheila, M., Massimo, P., Katia, S., Deborah, M., Evren, S. and Naveen, S., "Bringing Semantics to Web Services with OWL-S," *World Wide Web*, pp. 243-277, 2007.
- [5] Ranganathan, A. and Campbell, R., "A Middleware for Context-Aware Agents in Ubiquitous Computing Environments," *In ACM/IFIP/USENIX International Middleware Conference 2004*, 2004.
- [6] Fikes, R. and Nilsson, N., "Strips: A new approach to the application of theorem proving to problem solving," *In J. Allen, J. Hendler, and A. Tate, editors, Readings in Planning*, pp.88-97. Kaufmann, San Mateo, CA, 1990.
- [7] Smith, M., Welty, C., Deborah, L. and McGuinness, D., "OWL Web Ontology Language Guide," W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>.
- [8] The OWL Service Coalition. OWL-S 1.1 release. Available at <http://www.daml.org/services/owl-s/1.1/>, November 2004.
- [9] Christensen, E., Curbera, F., Meredith, G and Weerawarana, S., "Web Services Description Language (WSDL) 1.1," W3C Note, <http://www.w3.org/TR/wsdl>, 2001.
- [10] Sirin, E., Parsia, B., Wu, D., Hendler, J. and Nau, D., "HTN planning for Web Service composition using SHOP2," *Journal of Web Semantics*, 1(4), 2004.
- [11] Klusch, M. and Gerber, A., "Evaluation of Service



Composition Planning with OWLS-XPlan," *Proc. of the 2nd IEEE International Workshop on Service Composition (SerComp)*, IEEE CS Press, Hongkong, China, 2006.

- [12] "FOAF Vocabulary Specification," Namespace Document 27 July 2005, <http://xmlns.com/foaf/0.1/>.
- [13] John, P., "The Logical Foundations of Goal-Regression Planning in Autonomous Agents," *Artificial Intelligence*, pp.267-335, 1998.
- [14] Sirin, E., Parsia, B., Grau, C., Kalyanpur, A. and Katz, Y., "Pellet: a practical OWL-DL reasoner," *Submitted for publication to Journal of Web Semantics*, 2006.
- [15] HP Labs Semantic Web Programme, (February 2004), Jena: A Semantic Web Framework for Java, (Online), Available from: <http://jena.sourceforge.net/index.html>.
- [16] Evren, S., Bernardo Cuenca, G. and Bijan, P., "Optimizing Description Logic Reasoning for Nominals," In *Proceedings of KR'2006*, pp.90-99, 2006.



#### 최정화

2004년 2월 숭실대학교 정보과학대학 컴퓨터학부 졸업(학사). 2006년 2월 숭실대학교대학원 컴퓨터학과 졸업(석사). 2006년 3월~현재 숭실대학교대학원 컴퓨터학과 박사과정. 관심분야는 유비쿼터스 컴퓨팅, 시맨틱 웹, 온톨로지 추론, 시맨틱 Annotation, 다중 에이전트 시스템 등

#### 박영택

정보과학회논문지: 소프트웨어 및 응용 제 36 권 제 2 호 참조