

## 템플릿 객체 모델을 이용한 URC 서비스 컴포지션 시스템

강상승<sup>○</sup> 하영국 손주찬  
한국전자통신연구원 지능형로봇연구단  
{kss<sup>○</sup>, ygha, jcsohn}@etri.re.kr

### URC Service Composition System Using Template Object Model

Sang-Seung Kang<sup>○</sup> Young-Guk Ha Joo-Chan Sohn  
Intelligent Robot Research Division, ETRI

#### 요 약

지능적인 로봇 서비스를 위해 기존의 로봇 태스크 및 행위 기반의 컴포지션 개념이 유비쿼터스 공간 상에 존재하는 모든 활용 가능한 서비스를 포함하는 개념으로 확대되어야 할 필요성이 있다. 본 논문은 시맨틱 웹 서비스 환경에서 URC 서비스를 표현하는 메타 정보를 기술한 서비스 템플릿 객체 모델을 기반으로 자동으로 웹 서비스를 선택하고 구성하여 적절한 로봇 서비스 플랜을 생성해주는 URC 서비스 컴포지션 시스템을 제안한다. 제안한 시스템은 기존의 웹 서비스들과 다른 컴포넌트들을 조합하고 연결하기 위해 필요한 액티비티들로 구성된 일련의 프로세스인 서비스 플랜을 생성해주며, 생성된 서비스 플랜을 기반으로 서비스 실행 엔진을 통해 해당 서비스를 호출하여 실제 로봇 서비스를 제공하게 된다.

#### 1. 서 론

기존의 제한된 영역에서의 서비스 개념이 유비쿼터스 공간 상에 존재하는 모든 활용 가능한 서비스를 포함하는 개념으로 확대되어야 할 필요성이 있다. 이와 같이 이질적인 공간 상에 존재하는 수많은 서비스들을 하나의 서비스로 구성하고 실행하기 위해서는 이중 플랫폼 상에 탑재되어 있는 서비스 객체들을 지능적으로 통합할 수 있어야 한다. 웹 서비스 기술은 단일 웹 서비스에 대한 정의와 비즈니스 레지스트리에 등록된 웹 서비스만을 이용할 수 있는 제한적인 구조이기 때문에 구문적인 확장은 가능하지만 의미적인 확장이 불가능한 한계를 내포하고 있다. 이러한 웹 서비스의 발전 과정에서 도출된 문제점의 해결방안으로 시맨틱 웹 서비스가 제시되고 있는데, 이것은 시맨틱 웹과 에이전트 기술을 웹 서비스에 적용시켜 웹 서비스 기술의 한계를 극복하려는 노력이다. 따라서 시맨틱 웹 서비스 환경을 통해 온톨로지 언어로 기술된 서비스의 의미 정보를 기반으로 지능적이고 자율적인 서비스를 생성하여 제공할 필요가 있다. 이에 따라 최근 시맨틱 웹 서비스를 도입하여 자동화된 웹 서비스의 탐색, 합성, 실행, 모니터링을 가능하게 하려는 노력들이 이루어지고 있다.

본 논문은 시맨틱 웹 서비스 환경에서 각각의 서비스를 표현하는 메타 정보를 기술한 서비스 템플릿 객체 모델을 기반으로 자동으로 웹 서비스를 선택하고 구성하여 적절한 서비스 플랜을 생성하여 로봇 서비스를 제공해주는 시스템을 제안하였다. 제안한 시스템은 지능형 서비스 로봇인 Ubiquitous Robotic Companion (URC)를 대상으로 한다. URC는 언제 어디서나 나와 함께 하며 나에게 필요한

서비스를 제공하는 로봇으로 정의되며, 기존 로봇에 네트워크 개념을 추가하여 저가에 다양한 기능이나 서비스 제공이 가능하도록 하는 로봇이다.

시맨틱 웹 서비스 환경에서 URC 서비스 컴포지션은 쿼리 텍스트 처리 모듈로부터 서비스의 목표 및 프로파일에 포함된 수행 서비스에 대한 정보를 정형화된 형태로 입력받아 실행 가능한 형태의 서비스 프로세스를 생성해 주는 기술이다. 제안한 시스템은 기존의 웹 서비스들과 다른 컴포넌트들을 조합하고 연결하기 위해 필요한 액티비티들로 구성된 일련의 프로세스인 서비스 플랜을 생성해준다. 생성된 서비스 플랜을 기반으로 서비스 실행 엔진을 통해 해당 서비스를 호출하여 실제 로봇 서비스를 제공하게 된다. 제안한 시스템은 로봇 서비스를 위한 시맨틱 웹 서비스 플랫폼을 기반으로 하지만, 웹 표준을 따르기 때문에 일반적인 웹 서비스 플랫폼에서도 운용 가능하다.

#### 2. URC 서비스 컴포지션

URC 서비스 플랜을 실행가능한 구체적인 플랜으로 생성하는 과정에서 사용자의 명령 정보를 정형화한 Input, Output, Precondition, Effect (IOPE) 정보를 획득하여 적절한 서비스 템플릿을 선택하고, 템플릿 객체 모델을 기반으로 최적의 URC 서비스들을 탐색, 선택 및 조합하여 서비스 플랜 인스턴스를 생성하게 된다. URC 서비스 컴포지션에서는 기존의 로봇 태스크 및 행위 기반의 컴포지션 개념이 유비쿼터스 공간상에 존재하는 모든 활용 가능한 서비스를 포함하는 개념으로 확대되어야 하며, 여기에는 현재 인터넷 상에 존재하고 있는 서비스들이 포함된다.

다. 이와 같이 유비쿼터스 공간 상에 존재하는 수많은 서비스들을 하나의 서비스로 구성하고 실행하기 위해서는 이종 플랫폼 상에 탑재되어 있는 서비스 객체들을 지능적으로 통합 할 수 있는 기술이 필요하게 되며, 제안한 시스템은 차세대 분산 객체 기술인 웹 서비스와 차세대 웹 기술인 시맨틱 웹을 접목한 시맨틱 웹 서비스 기술을 기반으로 이를 수행하게 된다.

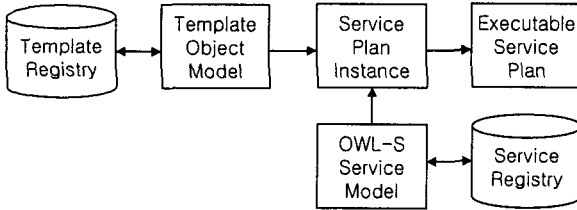


그림 1. URC 서비스 컴포지션 구조

제안한 시스템 구조는 그림 1과 같다. 템플릿 레지스트리로부터 적절한 템플릿을 선택하고 템플릿 객체 모델을 생성한다. 또한 서비스 레지스트리를 탐색하여 OWL-S 서비스 모델을 생성한다. 선택된 템플릿 객체 모델을 기반으로 추출한 OWL-S 단위 서비스를 구성하여 URC 서비스를 위한 서비스 플랜 인스턴스를 생성하고, 이를 실행가능한 서비스 플랜으로 변환하여 로봇 서비스를 제공하게 된다.

## 2.1 템플릿 객체 모델

서비스 템플릿은 서비스 컴포지션을 위해 사전에 정의된 기본 골격으로 각각의 서비스를 표현하는 메타 정보를 기술한 객체이다. 서비스 템플릿 레지스트리는 서비스 템플릿에 대한 시맨틱 기반의 인덱싱 서비스를 제공하며, 서비스 템플릿 리파지토리는 서비스 템플릿을 위한 저장 공간을 제공한다. 이러한 서비스 템플릿의 시맨틱에는 Input, Output, Precondition, Effect 등의 정보가 있으며, 각각의 정보를 표현하는 값들은 개념 온톨로지 및 서비스 온톨로지를 참조한다.

서비스 템플릿을 기술하기 위한 템플릿 객체 모델에서 Template 클래스는 BasicTemplate과 Aggregate Template의 두 가지 타입으로 구성된다. BasicTemplate은 최소 단위의 서비스 시맨틱을 기술하며, Aggregate Template은 하나 이상의 BasicTemplate 또는 AggregateTemplate의 집합으로 구성된다. 또한 Template 클래스는 관련 프로퍼티들로 hasParameter, hasInput, hasOutput, hasPrecondition 및 hasEffect를 가지며, 이들의 범위는 각각 Parameter, Input, ConditionalOutput, Precondition 및 ConditionalEffect 클래스의 서브클래스이어야 한다. Input은 프로세스가 실행을 위해 필요로 하는 정보이며, 프로세스의 실행 결과는 Output이다. Effect는 서비스의 시맨틱을 표현하는 핵심으로, Output이 단순 정보를 나타내는 반면 Effect는 실제 상황을 나타낸다. Precondition은 프로세스가 실행되기 위해 만족해야 하는 조건을 나타낸다. 템플릿의 제

어 구성자 정의를 위한 ControlConstruct 클래스는 Sequence, Choice, Split, Sync 등을 서브클래스로 가진다. Sequence는 컴포넌트의 일련된 연결을 표현하고, Choice는 다수의 컴포넌트 중에서 선택하게 하는 것을 나타내고, Split은 하나의 컴포넌트에서 다수의 컴포넌트로 분기 되는 것을 표현하며, Sync는 앞단의 컴포넌트들이 특정 시점에 동시에 끝나는 것을 나타낸다.

서비스 컴포지션을 위해서는 적절한 서비스 템플릿이 있어야 한다. 이 과정에서 요구되는 서비스 템플릿을 사용자 편의적인 환경에서 저작 가능하며 다양한 서비스 도메인 환경을 지원하는 저작 환경이 필요하다. 서비스 개발자는 다양한 서비스 템플릿을 저작하여 템플릿 레지스트리에 등록하여야 한다. 컨텍스트 처리 모듈로부터 받은 IOPE 정보의 시맨틱을 기반으로 템플릿 레지스트리를 검색하여 서비스 합성의 기반이 되는 적절한 서비스 템플릿을 선택하게 된다. 이후 선택된 서비스 템플릿으로부터 템플릿 객체 모델을 분석하고 단위 서비스들의 의미 정보를 분석하며, 서비스 플랜 인스턴스 생성에 필요한 기반 정보를 추출하는 기능을 수행하게 된다. 본 시스템에서 서비스 템플릿은 Web Ontology Language (OWL)을 기반으로 정의한 서비스 템플릿 기술 언어로 작성되었다. 다음은 하나의 서비스 템플릿 예의 일부분이다.

```

<tmpl:BasicTemplate rdf:ID="ReadSentenceTemplate">
  <rdfs:label>ReadSentenceTemplate</rdfs:label>
  <tmpl:representedBy>
    <tmpl:TemplateProfile
      rdf:ID="ReadSentenceTemplateProfile">
      <tmpl:hasInput
        rdf:resource="&HumanRelation:#Person"/>
      <tmpl:hasInput
        rdf:resource="&Misc:#Sentence"/>
      <tmpl:hasEffect
        rdf:resource="&Behavior:#Read"/>
      </tmpl:TemplateProfile>
    </tmpl:representedBy>
  </tmpl:BasicTemplate>
  
```

## 2.2 OWL-S 서비스 모델

적합한 서비스를 표현하고 있는 서비스 프로파일 정보를 검색하기 위해 서비스 레지스트리에 질의하고 응답받는 기능을 수행하며, OWL-QL 기반의 시맨틱 질의문을 생성 및 관리하는 기능을 포함한다. 여기서, 서비스 레지스트리는 서비스 프로파일에 대한 의미 정보를 기반으로 하여 인덱싱 서비스를 제공하는 소프트웨어 컴포넌트 역할을 한다. 제안한 시스템에서는 선택된 서비스 템플릿에 포함된 단위 서비스들의 의미 정보를 기반으로 서비스 레지스트리에 대한 서비스 탐색을 수행하게 되며, 서비스를 표현하고 있는 OWL-S 프로파일 정보와 웹 서비스 기술 정보 등을 포함하는 결과값을 가져오게 된다.

서비스 탐색을 통해 획득한 서비스의 의미정보는 OWL-S 서비스 모델로 작성된다. OWL-S는 서비스가 사용자나 다른 에이전트로부터 요구하는 것은 무엇이고 그

들에게 제공하는 것은 무엇인가를 나타내는 Profile, 서비스가 어떻게 작동하는가를 나타내는 Process, 서비스가 어떻게 사용되는가를 나타내는 Grounding으로 구성된다. 핵심 구성요소로서 서비스를 표현하는 온톨로지 클래스인 OWL-S 프로파일은 서비스의 의미 정보 표현을 위해 IOPE 정보를 이용한다. 다음은 하나의 OWL-S 서비스 및 프로파일 예의 일부분이다.

```
<service:Service rdf:ID="E30ReadSentenceService">
  <service:presents rdf:resource=
"#E30ReadSentenceServiceProfile"/>
  <service:describedBy rdf:resource=
"#E30ReadSentenceServiceProcessModel"/>
  <service:supports rdf:resource=
"#E30ReadSentenceServiceGrounding"/>
</service:Service>
<profile:Profile rdf:ID="E30ReadSentenceServiceProfile">
  <service:presentedBy rdf:resource=
"#E30ReadSentenceService"/>
  <profile:has_process rdf:resource=
"#E30ReadSentenceServiceProcessModel"/>
  <profile:serviceName>E30ReadSentenceService
Agent</profile:serviceName>
  <profile:hasInput rdf:resource="#Person"/>
  <profile:hasInput rdf:resource="#Sentence"/>
  <profile:hasEffect rdf:resource="#Behavior:#Read"/>
</profile:Profile>
<process:ProcessModel
rdf:ID="E30ReadSentenceServiceProcessModel">...
</process:ProcessModel>
<grounding:WsdIGrounding
rdf:ID="E30ReadSentenceServiceGrounding">...
</grounding:WsdIGrounding>
```

이후 서비스 탐색을 통해 획득한 서비스 정보들로부터 최적의 단위 서비스들을 추출하고, 추출한 서비스를 기반으로 OWL-S 단위 서비스 모델을 분석하고 플랜 인스턴스의 구성 단위 객체 모델을 생성하는 기능을 수행한다.

### 2.3 서비스 플랜 인스턴스

서비스 플랜 인스턴스는 메타 시맨틱을 기술하는 서비스 템플릿을 기반으로 실존하는 구체적인 서비스 객체를 바인딩하기 위한 정보를 기술한 것으로, 템플릿 선택 모델에서 선택된 서비스 템플릿을 기반으로 서비스 플랜 인스턴스를 생성하고, 생성된 서비스 플랜 인스턴스에 서비스 탐색 모델을 통해 선택된 단위 서비스들을 조합하여 사용자 명령을 처리하기에 적합한 서비스 플랜을 구성하게 된다. 다음은 상기 로봇 서비스 템플릿에 대해 생성된 서비스 플랜 인스턴스 예의 일부분이다.

```
<inst:TemplateInstance rdf:ID="ReadSentenceTemplat
eInstance">
  <inst:instantiates rdf:resource="#ReadSentenceTem
```

```
plate:#ReadSentenceTemplate"/>
  <inst:hasGroundings rdf:parseType="Collection">
    <inst:Grounding>
      <inst:forTemplate rdf:resource="#ReadSentenceT
emplate:#ReadSentenceTemplate"/>
      <inst:byService rdf:resource="#E30ReadSentence
Service:#E30ReadSentenceServiceGrounding"/>
    </inst:Grounding>
  </inst:hasGroundings>
</inst:TemplateInstance>
```

상기의 과정을 통해 생성된 서비스 플랜은 서비스의 접근 방법을 구체화한 OWL-S 그라운드링 정보와 웹 서비스를 기술한 WSDL 정보를 기반으로 실행 가능한 서비스 플랜으로 변환하고, 서비스 실행 엔진에 적합한 실행 프로세스 혹은 실행 스크립트를 생성하며, 상기의 프로세스나 스크립트를 로봇 서비스 호출 시스템을 통해 실행함으로써 사용자에게 실제 서비스를 제공하게 된다.

### 3. 결 론

유비쿼터스 공간 상에 존재하는 수많은 서비스들을 하나의 서비스로 구성하고 실행하기 위해서는 이중 플랫폼 상에 탑재되어 있는 서비스 객체들을 지능적으로 통합할 수 있어야 한다. 본 논문은 서비스의 시맨틱 정보를 기반으로 서비스 에이전트의 지능적이고 자율적인 서비스 제공을 위해 제시된 시맨틱 웹 서비스 환경을 기반으로 URC 서비스 컴포지션 시스템을 제안하였다. 제안한 시스템은 URC 서비스를 표현하는 메타 정보를 기술한 서비스 템플릿 객체 모델을 기반으로 자동으로 웹 서비스를 선택하고 구성하여 적절한 로봇 서비스 플랜을 생성해준다. 제안한 시스템은 기존의 웹 서비스들과 다른 컴포넌트들을 조합하고 연결하기 위해 필요한 액티비티들로 구성된 일련의 프로세스인 서비스 플랜을 생성하며, 생성된 서비스 플랜을 기반으로 서비스 실행 엔진을 통해 로봇 서비스를 제공하는 특징을 가진다.

### 참고문헌

- [1] S. McIlraith, T. C. Son, and H. Zeng, "Semantic Web Service," *IEEE Intelligent Systems*, 16(2):46-53, 2001.
- [2] 조영조, "지능형 서비스 로봇과 URC", *주간기술동향*, 통권 1150호, 2004.
- [3] S. Kang and J. Sohn, "A Framework for Service Planning in the Semantic Web Services," *ITC-CSCC 2005*, vol.2, pp.727-8, 2005.
- [4] The OWL Services Coalition, OWL-S: Semantic Markup for Web Services, <http://www.daml.org/services/owl-s/1.0/owl-s.html>
- [5] D. Martin, M. Burstein, O. Lassila, M. Paolucci, T. Payne, and S. McIlraith, Describing Web Services using OWL-S and WSDL, <http://www.daml.org/services/owl-s/1.0/owl-s-wsdl.html>