

웹 서비스 기반 실행 프로세스 생성에 관한 연구

박천수, 손주찬
한국전자통신연구원, 지능형로봇연구
e-mail : bettle@etri.re.kr

A Study on Executable Process Generation based on Web Service

Cheonshu Park and Joochan Sohn
Intelligent Robot Research Division, ETRI

요 약

본 논문은 지능형 로봇을 통하여 사용자가 원하는 서비스를 제공 받기 위해 외부의 웹 리소스를 이용하여 최적의 서비스 컴포지션 과정을 거쳐 실행 가능한 형태의 언어로 생성하는 방법을 제시한다. 온톨로지 형태로 정의된 템플릿을 서비스 컴포지션을 통해 플랜 인스턴스를 생성하고, 구축된 웹 서비스와 온톨로지를 이용하여 서비스 플랜에 맞게 실행 가능한 형태의 언어인 BPEL4WS 를 생성 한다. 이를 통하여 기존에 제공되었던 제한적이고 수동적인 서비스를 외부의 웹 서비스를 이용하여 보다 많은 정보를 지능형 로봇을 통해 제공 할 수 있다.

1. 서론

기존의 로봇은 하나의 작업을 수행 하기 위해 필요한 정보를 로봇에 내장된 서비스만을 이용하여 수행 할 수 있는 형태의 수동적이고 고립된 형태의 로봇인데 반해, 본 연구는 능동적이고 자율적인 로봇을 구현 하려는 목표를 가지고 네트워크를 이용한 언제 어디서나 나와 함께 하며 나에게 필요한 서비스를 제공하는 지능형 서비스 로봇인 URC(Ubiquitous Robotic Companion) 개념을 기반으로 한다[1].

본 연구에서 제안한 서비스 모델은 URC 개념을 기반으로 하고 있으며, 자율적이고 능동적인 서비스를 제공하는 로봇을 구현하기 위한 하나의 방법을 제시 한다. 본 논문은 지능형 로봇을 통해 수행 할 서비스를 OWL-S[5] 기반의 템플릿 형태로 정의 하고, 정의된 여러 개의 서비스들 중에서 서비스 컴포지션을 통해 최적의 서비스 플랜을 생성한다. 본 시스템은 생성된 서비스 플랜 인스턴스를 입력으로 받아, 실행 엔진에서 실행 할 수 있는 형태의 언어인 BPEL4WS[4]를 생성한다. 이렇게 생성된 BPEL4WS 는 프로세스를 실행 할 수 있는 실행 엔진을 통해 생성된 프로세스의 흐름에 맞게 실행된다. 즉, 서비스를 제공하기 위

해 AXIS[6]엔진을 통해 미리 배포된 웹 서비스[2], [3]를 이용하여 실행 엔진은 생성된 BPEL4WS 의 프로세스에 따라 웹 서비스를 실행 한다.

2. 배경

2.1 웹 서비스

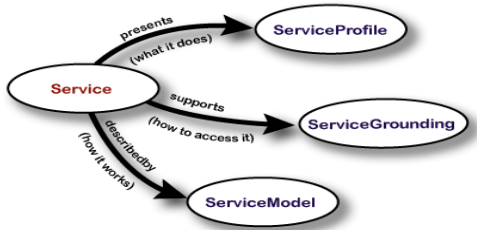
웹 서비스는 표준화된 SOAP 메시지를 통하여 접근 가능한 네트워크 명령들의 집합을 기술한 인터페이스이다. 웹 서비스는 메시지 포맷, 전송 프로토콜, 위치 등을 포함하고 서비스를 위해 필요한 요소들을 XML 을 통해 표현하고, 인터페이스를 통해 서비스 구현의 세부사항을 숨김으로써, 서비스가 구현된 하드웨어나 소프트웨어 플랫폼에 독립적이고, 또한 구현된 프로그래밍 언어에 독립적으로 사용될 수 있는 특징을 가지고 있다.

2.2 OWL-S

자동화된 웹 서비스 컴포지션을 위해서는 기본적으로 각각의 웹 서비스에 대한 의미 기술 방법이 필요하다. OWL 을 기반으로 웹 서비스를 기술하기 위해 작성된 OWL-S 는 웹 서비스의 의미를 정의하는 상위

온톨로지이다. 즉, 각각의 웹 서비스는 OWL-S 온톨로지의 인스턴스가 되며, 서비스 에이전트는 이러한 서비스 인스턴스에 대한 추론을 통하여 기본적으로 다음과 같은 웹 서비스 태스크를 수행할 수 있다.

- 자동화된 웹 서비스 검색 (Automatic Web Service Discovery)
- 자동화된 웹 서비스 실행 (Automatic Web Service Invocation)
- 자동화된 웹 서비스 컴포지션 및 상호 운용성 (Automatic Web Service Composition and Interoperation)



<그림 1> Top level of the service ontology

OWL-S 는 <그림 1>과 같이 웹 서비스를 정의하는 Service 온톨로지, 서비스가 무엇을 하는지(What it does)를 표현하는 Service Profile 온톨로지, 서비스의 프로세스를 구체적으로 기술하는(How it works) Service Model 온톨로지, 마지막으로 서비스의 접근 방법을 기술하는(How to access it) Service Grounding 온톨로지로 구성된다.

2.3 BPEL4WS (Business Process Execution Lang for Web Service)

BPEL4WS 는 웹 서비스를 위한 프로세스를 모델링 하기 위한 언어로서 프로세스와 프로세스 파트너간의 상호작용을 통한 비즈니스 프로세스의 행위를 기술하기 위한 모델과 문법을 정의 한다. 즉, 웹 서비스를 포함하여 비즈니스 프로세스를 명시하기 위한 특화된 언어이다.

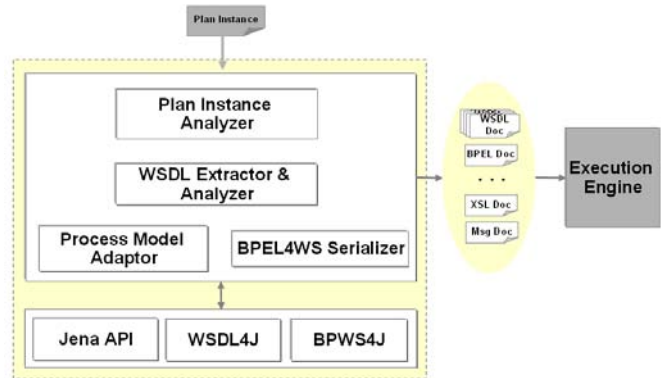
BPEL4WS 는 프로세스에 실행에 대한 내부적인 행위(절차)를 숨김으로써 비즈니스 프로토콜을 기술하는데 유용하고, 프로세스에 대한 실제 행위를 모델링 함으로써 직접 호출 가능한 서비스로 컴파일 될 수 있다. 본 논문에서는 비즈니스 목적을 위한 사용 보다는 웹 서비스를 실행하기 위한 실행 엔진 기술로 사용하였으며, 이를 위해 템플릿 기반의 로봇 실행 제어를 위한 서비스 플랜 인스턴스를 BPEL4WS 로 변환 하게 된다.

3. 시스템 설계

<그림 2>은 웹 서비스를 실행 하기 위한 실행 플랜 생성 모듈에 대한 전체적인 시스템 구조로 템플릿 기반의 서비스 플랜을 이용하여 필요한 서비스 컴포지션 과정을 통해 생성된 Plan Instance 를 입력으로 받아서, 최종적으로 BPEL4WS 문서를 생성해 줌으로써,

웹 서비스를 통해 로봇에 서비스를 제공하고 제어할 수 있는 구조를 가진다.

입력 받은 Plan Instance 를 분석하여 서비스의 흐름(Flow)와 해당 서비스를 찾을 수 있도록 Plan Instance 분석기에서 웹 서비스의 물리적인 위치를 찾아 내고, 서비스 흐름을 BPEL4WS 의 서비스 흐름과 매핑 시키기 위해 미리 정의한 Template Instance 를 참조한다.



<그림 2> 시스템 구조

Plan Instance 에 기술된 물리적인 웹 서비스의 정보를 제공하는 Grounding 온톨로지를 통해 WSDL 문서를 찾고, WSDL 에서 제공하는 Port Type 을 이용하여 WSDL 에서 제공되는 Operation 들을 참조 한다. BPEL 문서를 생성하기 위해 OWL-S 형식으로 작성된 온톨로지를 로딩하여 실행을 위해 필요한 요소들을 추출 한다.

WSDL 에 정의된 메시지를 참고하여 BPEL 에서의 Input, Output 메시지 형식을 작성하고, 수행 절차를 기술하는 Flow 를 추출하여 모델 변환 모듈을 통해 BPEL 실행에 필요한 형태의 메모리 상의 모델로 변환한다.

메모리 상의 모델을 Serializer 를 통해 파일로 생성하게 되며 생성된 문서를 검증 한 후, 실행 엔진 모듈로 전송한다.

3.1 플랜 인스턴스 분석기(Plan Instance Analyzer)

Plan Instance 에 대한 정보를 로딩하여 웹 서비스를 실행 하기 위해 필요한 WSDL 문서에 대한 정보를 분석하는 모듈로 우선, Plan Instance 내에 있는 웹 서비스에 대한 Grounding information 을 추출하여 grounding 온톨로지를 참조한다. Grounding 온톨로지 내에 있는 여러 개의 Atomic process 를 가져와 각각의 message 와 operation 을 정의하고 있는 WSDL 의 URI 를 가져온다.

3.2 WSDL 추출 및 분석기(WSDL Extractor & Analyzer)

WSDL 추출 및 분석기는 Plan Instance 모듈에서 가져온 WSDL 의 URL 을 로딩 하여 BPEL4WS 을 생성하기 위해 필요한 Message type, Operations, Port type, Service 등에 대한 정보를 분석하고 추출하는 기능을 담당한다.

WSDL 문서는 definition 요소를 root 로 하여 내부에

웹 서비스의 명세를 정리한 types, message, portType, binding, service 요소들을 포함하여 구성된다. WSDL의 문서에 포함된 웹 서비스의 정의는 오퍼레이션의 종류와 각 오퍼레이션들의 파라미터, 데이터 타입 정의 등의 추상적인 부분과 전송 프로토콜, Serialization, End-point 등의 서비스를 제공하는 방식에 대한 정의를 포함하는 유형 정의의 부분으로 나뉜다. 즉, 특정 오퍼레이션이 어떤 내용의 메시지들로 요청되고, 응답되는지에 대한 추상적인 설명이 추상정의에서 이루어지고, 이렇게 정의된 웹 서비스 오퍼레이션들에 어떤 인코딩 방식을 사용하여 실제 전송 메시지를 만들어 어떤 전송 방식으로 전송할 것인지, 실제 요청을 받아들일 수 있는 곳인 end-point에 대한 정의를 유형 정의의 부분에서 하게 된다.

WSDL 문서에서 Binding 요소는 portType 요소에서 정의된 오퍼레이션, 메시지들에 대한 오퍼레이션을 실행/구축하기 위한 메세징 프로토콜, 전송 프로토콜 등을 확정하는 역할을 한다. 주로 메세징 프로토콜로는 SOAP, 전송프로토콜로는 HTTP를 주로 이용한다. Service 요소는 서비스의 이름과 엔드 포인트를 지정한다.

3.3 프로세스 모델 변환기(Process Model Adaptor)

프로세스 모델 변환기는 템플릿을 기반으로 생성된 플랜 인스턴스(Plan Instance)에 대한 모델을 BPEL4WS를 생성하기 위해 변환하는 모듈로 CompositeInstanceManager에서 받아 들인 플랜 인스턴스(Plan Instance)를 로딩하여 서비스 템플릿 인스턴스(Service Template Instance)에 정의된 서비스 프로세스(Activity)와 WSDL에 정의된 message, portType, service, operation 등을 이용하여 BPEL 형태의 모델을 생성한다. WSDL에서 정의된 In/Out message types은 BPEL에서 Variable로 매핑이 되고, 서비스 템플릿 인스턴스에서 정의된 Control constructor들은 BPEL에서 Activity로 매핑 된다.

웹 서비스에서 정의된 오퍼레이션의 실행 결과를 이용하기 위해 reply 요소를 이용하여 서비스 실행 결과를 얻을 수 있도록 정의 한다. 즉, 템플릿 기반으로 생성된 플랜 인스턴스(Plan Instance)문서를 로딩하여 얻은 정보를 BPEL4WS 명세에 정의된 요소들을 매핑을 통해 변환하는 모듈이다.

3.4 BPEL4WS Serializer

BPEL4WS Serializer 모듈은 OWL-S로 정의된 온톨로지를 기반으로 생성된 플랜 인스턴스를 입력으로 받아 프로세스의 실행에 필요한 요소들을 추출하여 하나의 실행 가능한 BPEL 모델을 생성하게 되고, 이렇게 생성된 메모리상의 BPEL 모델을 Serializer를 통해 BPEL4WS 문서를 생성하게 되고, 생성된 문서를 검증하는 모듈이다.

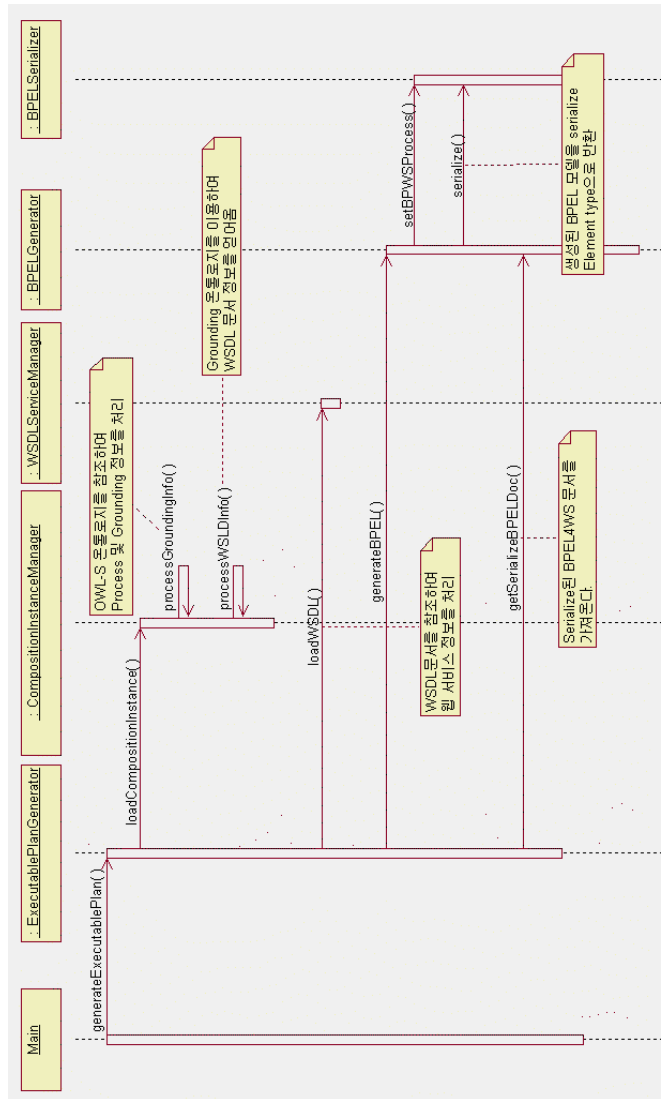
즉, OWL API를 이용하여 분석된 플랜 인스턴스의 정보를 토대로 BPEL의 Activity를 생성한다. Activity에서 사용되는 웹 서비스들의 정보를 “WSDL 분석 및

추출기”에서 추출하여 “모델 변환기”에서 매핑 규칙을 이용하여 수행 가능한 형태의 BPEL 모델을 생성하고 생성된 모델은 Serializer를 통해 BPEL 문서로 생성된다.

3.5 실행 프로세스 생성

우선, 사용자의 명령어를 받아들여 사용자 질의 분석을 통해 음성을 로봇이 이해할 수 있는 언어로 변환한다. 변환된 언어를 이용해 Context 처리 및 서비스 제어 모듈에서 현재의 상황 정보를 판단하여 서비스에 대한 목표를 결정한다. 생성된 목표를 이용하여 사용자가 원하는 최적의 서비스를 제공하기 위해 정의된 서비스 템플릿과 배포된 웹 서비스를 검색하여 찾아낸 후, 서비스 실행 플랜을 생성한다. 서비스 실행 플랜을 입력으로 받아 실행 엔진은 실행 모델을 생성하고, 해당 웹 서비스를 찾아 실행하게 된다.

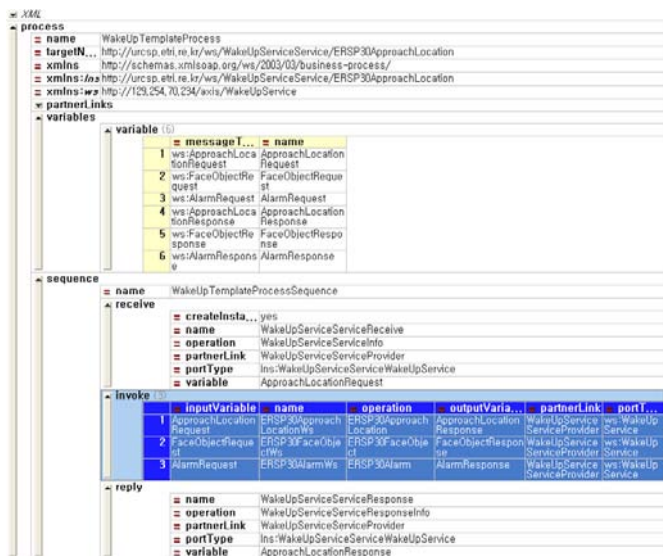
<그림 3>은 플랜 인스턴스를 입력으로 받아서 최종적으로 BPEL4WS 문서를 생성하기까지의 흐름을 나타낸 주요 클래스의 시퀀스 다이어그램이다.



<그림 3> 실행 플랜 생성 시퀀스 다이어그램

생성 과정을 살펴보면 다음과 같다.

1. 메인 모듈에서 실행 가능한 형태의 문서를 생성하기 위해 ExecutablePlanGenerator 클래스의 generateExecutablePlan() 메소드를 호출한다. 이때, 템플릿을 기반으로 생성된 Composition Instance 모델을 넘겨준다.
2. ExecutablePlanGenerator 클래스에서 Composition Instance 를 분석하기 위해 CompositionInstanceManager 클래스의 loadCompositionInstance() 메소드를 호출한다. CompositionInstanceManager 클래스에서는 OWL-S 로 기술된 서비스를 구성하는 프로세스를 분석하고, WSDL 문서에 대한 정보를 Grounding 온톨로지로부터 추출하여 한다.
3. 추출된 Grounding 온톨로지 정보를 이용하여 WSDLSERVICEManager 는 WSDL 문서를 참조하여 웹 서비스에서 제공되는 message, portType, operation, service 등에 대한 정보를 추출하여 BPEL 을 생성하기 위한 모듈로 전송한다.
4. BPELGenerator 클래스는 ExecutablePlanGenerator 에서 가져온 정보를 이용하여 템플릿 온톨로지 에서 제공하는 ControlConstructs 와 BPEL 의 activity 들을 매핑시키고, 서비스 Invoke 를 위해 필요한 WSDL 의 정보를 이용해 BPEL 모델을 생성한다.
5. 생성된 BPEL 모델은 Serializer 를 통해 BPEL4WS 문서로 생성 되고, 검증을 통하여 실행 엔진에 입력값으로 전송한다.



<그림 4>Wake up 관련 BPEL4WS 생성 결과

<그림 4>는 템플릿 기반의 서비스 플랜 인스턴스를 입력 받아 최종적으로 생성된 Wake-up 관련 BPEL4WS 문서를 나타낸다. 생성된 BPEL4WS 문서를 JMS(Java Message Server)를 통해 URC Process Execution Engine 에

전달 한다. 이때, 실행 엔진에서 필요한 WSDL 문서들의 URI, 최종 결과를 얻기 위한 XSL 문서, 입력 변수 값, Transaction ID, Session ID 등을 함께 전달 한다. 실행 엔진은 생성된 프로세스를 실행하여 결과를 넘겨 준다.

4. 결론

본 연구는 URC 지능형 웹 기반의 로봇 서비스 제공을 위한 플랫폼 기술 개발의 한 부분으로 URC 서비스 컴포지션 기술의 실행 프로세스 생성 모듈을 구현하였다.

본 논문에서는 사용자의 명령으로부터 서비스 컴포지션을 거쳐 수행할 일련의 서비스 실행 플랜을 구성하고, 이를 기반으로 프로세스 실행 엔진을 통해 실행 가능한 형태인 BPEL4WS 를 생성하는 방법을 제시 하였다. 이를 통해, 기존에 제공되었던 제한적이고 수동적인 서비스를 외부의 웹 서비스를 이용하여 보다 다양한 형태의 서비스를 지능형 서비스 로봇을 통해 제공할 수 있다.

참고문헌

- [1] URC (Ubiquitous Robotic Companion), “IT 차세대 성장동력 기획보고서(지능형서비스로봇),” 2003. 12.
- [2] W3C, Web Service Description Working Group, <http://www.w3.org/2002/ws/desc/>
- [3] W3, Web Service Activity, <http://www.w3.org/2002/ws/>
- [4] T. Andrews, et al, “BPEL4WS: Business Process Execution Language for Web Services Version 1.1”, <http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>, 2003.
- [5] OWL-S Home Page, “OWL-S: Semantic Markup for Web Services,” <http://www.daml.org/services/owl-s/1.1/overview/>, 2004.
- [6] Apache, “AXIS”, <http://ws.apache.org/axis/>