

A Model-Based Interface to Cloud Services for Intelligent Service Robots

Byunggi Choi[†] · Jonguk Lee^{††} · Sunggi Park^{†††} · Jaeho Lee^{††††}

ABSTRACT

Service robots providing services according to user's needs in dynamically changing environments should be able to utilize external services such as cloud services. Cloud services are, however, changing and expanding continuously and thus the interface to the services must be general and flexible to adapt to the changes of the functionality and data of the services. In order to facilitate the adaptation of the interface to the changes, a model-based general interface to various cloud services is proposed. In this approach, a general and extensible interface is realized by defining standard service profiles that can be easily extended to adapt to the changed services. Experiments with intelligent service robots show satisfying results exhibiting flexible adaptations to new or changed external services.

Keywords : Service Robot, Cloud Interface, Model-based Interface

지능형 서비스 로봇을 위한 모델 기반 클라우드 서비스 인터페이스

최 병 기[†] · 이 중 옥^{††} · 박 성 기^{†††} · 이 재 호^{††††}

요 약

유동적으로 변화하는 환경에서 사용자의 요구에 따라 서비스를 제공하는 서비스 로봇은 클라우드 서비스와 같은 외부 서비스를 이용할 수 있어야 한다. 그러나 클라우드 서비스는 지속해서 변화하고 확장되므로 서비스의 기능 및 데이터 변경에 적응할 수 있는 일반적이고 유연한 인터페이스가 필요하다. 이를 위해 다양한 클라우드 서비스에 대한 모델 기반의 일반적인 인터페이스를 활용하여야 한다. 이러한 접근 방법을 통해 변경된 서비스에 접근할 수 있는 표준 서비스 프로파일을 정의함으로써 일반적이고 확장 가능한 인터페이스를 구현하였다. 또한, 지능형 서비스 로봇을 활용한 실험을 통해 새로운, 혹은 변경된 외부 서비스에 대해 유연하게 적응하는 결과를 검증하였다.

키워드 : 서비스 로봇, 클라우드 인터페이스, 모델 기반 인터페이스

1. 서 론

지능형 로봇과 같이 범용적 서비스를 수행하여야 하는 시스템은 개방적, 확장적으로 외부 서비스와 연계함으로써 수행할 수 있는 서비스의 범위를 넓힐 수 있다. 여기서 지능형 로봇이란 다양하고 유동적으로 변화하는 환경 내에서 사용자의 요구에 적합한 행동을 수행할 수 있는 로봇을 의미하고 [1], 범용적 서비스란 로봇의 실행 대상인 서비스가 단순히 하나의 서비스만을 제공하는 것이 아니라 사용자의 요구에

따라 포괄적이고 광범위한 서비스를 실행할 수 있음을 의미하며, 외부 서비스란 Service Oriented Architecture[2]에서 말하는 서비스와 같은 형태로 Cloud 상에 존재하는 SaaS (Software as a Service) 기반 소프트웨어 서비스를 말한다.

그러나 클라우드 서비스는 지속해서 새로운 서비스가 개발, 개선되고 있어 이에 대응하기 위해 서비스의 기능 및 데이터 변경에 적응할 수 있는 일반적이고 유연한 인터페이스가 필요하고, 클라우드 서비스는 서비스의 종류에 따라 필요한 Parameter의 종류나 접근 방법 등이 달라지는데, 지능형 플랫폼인 서비스 로봇에서 개방적인 서비스 접근을 가능하게 하기 위해서는 다음과 같은 요소가 필요하다.

1.1 서비스 모델링을 통한 모델 기반 인터페이스

외부 소프트웨어 서비스는 다양한 종류의 접근 방법과 데이터 형식을 가지고 있으며, 접근하고자 하는 대상 서비스가 변경되더라도 시스템의 변경 없이 일관된 인터페이스로 연계하기 위해서는 외부 서비스를 활용하는 데 적합한 추상화된

※ 본 연구는 산업통상자원부의 재원으로 기술혁신사업의 지원을 받아 수행한 연구 과제(No. 10060086, 개인 서비스용 로봇을 위한 지능-지식 집약·개방·진화형 로봇지능 소프트웨어 프레임워크 기술 개발)입니다.
[†] 정 회 원 : 서울시립대학교 전자전기컴퓨터공학과 박사과정
^{††} 정 회 원 : 서울시립대학교 정보기술연구소 연구원
^{†††} 비 회 원 : 한국과학기술연구원 지능로봇연구단 책임연구원
^{††††} 종신회원 : 서울시립대학교 전자전기컴퓨터공학부 교수
Manuscript Received : August 5, 2019
First Revision : October 4, 2019
Second Revision : November 28, 2019
Accepted : December 13, 2019
* Corresponding Author : Jaeho Lee(jaeho@uos.ac.kr)

모델[3]이 필요하다. 모델은 외부 서비스의 인증 방법과 서비스의 종류에 따라 확장성과 편의성을 확보해야 하며, 시스템은 모델의 교체 및 추가만으로 연계 가능한 서비스를 확장할 수 있어야 한다.

1.2 소프트웨어 서비스의 표준적인 명세

외부 서비스는 그 종류에 따라 각각이 필요로 하는 파라미터가 상이하다. 서비스의 주소, 서비스에서 사용하는 파라미터들의 단위나, 사용되는 파라미터의 형식, 파라미터가 배치되는 위치, 서비스의 인증방식 등 각각의 서비스별로 필요로 하는 파라미터를 서술하기 위한 표준적인 서비스 서술 방법이 필요하다.

1.3 시스템과 서비스 간의 연계를 위한 파라미터 맵핑

외부 서비스를 이용해 획득한 데이터를 통합 대상이 될 시스템에서 사용하기 위해서는 서비스에서 일차적으로 획득한 데이터가 아닌 통합 대상에서 이용할 수 있는 형식의 데이터 변환 및 맵핑을 수행할 수 있어야 한다.

이와 같은 요구사항을 만족하기 위하여 본 논문에서는 지식 모델을 통해 확장 가능한 서비스 인터페이스를 구현할 수 있는 모델 기반 클라우드 서비스 연계 방법에 대해 연구하고 이를 구현 및 검증하였다. 해당 연계 방법은 모델을 바탕으로 클라우드 서비스 실행을 요청하거나 알림을 받을 수 있는 인터페이스 컴포넌트를 생성할 수 있으며, 이를 통해 로봇과 같이 범용 작업을 수행하여야 하는 시스템이 자동으로 외부 서비스와 연계하는 방법을 제공한다.

2. 관련 연구

서비스 로봇의 모델 기반 클라우드 서비스 연계를 위해서는 클라우드 서비스에 필요한 인자와 사용되는 리소스, 그리고 표준적인 인증 절차를 수행할 수 있는 인증 정보가 서술된 명세가 필요하며, 이러한 명세를 바탕으로 작성된 서비스 모델과 서비스 모델을 바탕으로 다른 시스템에서 활용 가능한 형식이나 통신방법으로 가공, 연계할 수단이 필요하다. 그리고 마지막으로, 연동 가능한 클라우드 서비스 모델을 실시간으로 추가할 수 있어야 한다.

클라우드 서비스의 모델 기반 접근 방법은 서비스 로봇 연계 목적만이 아니라 여러 분야에서 연구되었는데, 이 중 RAML[4]은 웹상에 게재되는 클라우드 서비스를 서술하고 이를 바탕으로 클라우드 서비스를 검증, 더 나아가 구축하기 위한 모델링 언어이며, Niklas Scholz가 제안한 RESTful api를 위한 모델 기반 시스템[5]은 웹상의 Single Page Application에서 여러 종류의 REST Api 사용 시 발생할 수 있는 문제점을 해결하려는 방안으로 모델 기반 접근법을 활용한 인터페이스 및 시스템 구조에 대해 제안하였다. 마지막으로 Web Service[6]는 웹상에서 제공되는 애플리케이션 서비스를 위한 명세와 프로토

콜, 통신 절차와 리소스 명세를 제공한다.

2.1 RAML

RAML(Restful API Modeling Language)은 마크업 언어인 YAML(Yet Another Modeling Language)[7]를 사용하여 REST 아키텍처를 기반으로 하는 클라우드 서비스의 명세와 이를 바탕으로 한 서비스 모델링을 지원하는 모델링 언어이다.

RAML을 활용한 서비스 명세에는 클라우드 서비스에 사용되는 리소스와 인증 정보, 인자값의 타입이나 필수 인자 여부 등을 서술할 수 있으며, 이를 바탕으로 프로토타입 서버를 구축하여 클라우드 서비스의 API 명세를 쉽게 검증할 수 있다.

2.2 Niklas Scholz의 인터페이스

Niklas Scholz는 Single Page Application에서 모델 기반으로 클라우드 서비스를 활용할 수 있는 인터페이스를 제안하였다. 이는 최근 마이크로서비스 위주로 작성되는 단일 페이지 애플리케이션(Single-page application)에서 많은 수의 클라우드 API를 연계하기 위해 제안되었으며, Niklas는 해당 방식을 통해 싱글 페이지 애플리케이션 개발 시 여러 종류의 API를 사용할 때와 이종의 API 간의 데이터 변환 과정에서 발생하는 개발 단계에서의 복잡성을 해소하려 하였다.

해당 인터페이스는 클라이언트가 Query Service 서버에 서비스를 요청하면, Query Service 서버가 GraphQL을 사용하여 데이터 변환과 RESTful Api 요청/응답 과정을 수행하여 최종 실행 결과를 클라이언트에게 반환한다. 이 과정에서 Query Service 서버는 사전에 작성된 요청 변환 함수와 GraphQL 스키마 모델을 사용하여 간략화된 요청 정보를 클라우드 서비스 수행에 적합한 형식으로 변환한다.

2.3 Web Service

웹 서비스는 기존의 웹 프로토콜과 Open XML 표준에 기반을 두어 애플리케이션 간 상호작용을 위한 체계적이고 확장 가능한 프레임워크를 제공하기 위해 제안되었다.

웹 서비스는 웹 서비스 간의 통신을 지원하기 위한 XML 기반 경량 프로토콜인 SOAP(Simple Object Access Protocol)[8]과 웹상에서 존재하는 서비스의 발견 및 사용을 위한 비즈니스, 조직 단위의 서비스 레지스트리에 대해 서술하는 UDDI(Universal Description, Discovery and Integration)[9]와 서비스의 접근 방식, 프로토콜 구조, 요청 및 응답 등에 관해 서술하는 WSDL(Web Service Description Language)[10]로 구성되어 있다.

Web Service를 구성하는 WSDL에서는 웹 서비스를 사용하기 위한 상세 명세를 제공한다. 이는 웹 서비스의 기능에 대한 설명과 서비스가 사용하는 리소스에 대한 구조, HTTP 요청과 응답(Request and Response)에 사용되는 데이터, 여러 발생시 응답과 HTTP, SOAP 등의 다른 프로토콜을 사용할 경우의 연결 방법 등에 대한 정보를 전부 포함한다.

2.4 시사점

RAML은 클라우드 서비스의 서버 구축을 지원하기 위해 클라우드 서비스의 상세한 명세와 이를 바탕으로 한 프로토타입 서버 구축 기능, API 요청에 대한 유효성 검증 기능 등을 포함하지만 클라이언트 측면에서의 서비스 활용을 위한 데이터 가공, 연계 기능이 포함되어 있지 않고, Niklas Scholz의 모델 기반 접근 방법은 웹상에서 복수의 클라우드 API를 사용할 때 발생하는 개발의 어려움을 해소하기 위해 GraphQL을 활용하여 서비스 간 인터페이스 기능을 지원하지만, 해당 인터페이스 구조는 서비스 모델의 실시간 추가가 불가능하고 표준적인 인증 절차에 대한 명세가 존재하지 않아 클라우드 서비스에 대한 자동적인 연계를 수행할 수 없다. Web Service는 웹상에 게재되는 서비스에 대한 인터페이스, 명세 및 사용되는 리소스에 대한 정보를 전부 제공하지만, 최근의 클라우드 서비스에서 사용되는 표준적인 인증방식인 OAuth나 API key 방식의 서비스를 서술할 수 없다.

따라서 본 논문에서는 이러한 기존 연구들의 한계를 극복하기 위해 서비스 자체에 대한 모델링과 서비스 모델의 실시간 추가, OAuth나 API key 등의 표준적인 인증방식 명세, 서비스 로봇에 적합한 모델 기반 프로토콜 맵핑 절차가 포함된 클라우드 서비스 인터페이스를 제안한다.

3. 배경 연구

지능형 서비스 로봇은 변화하는 환경에서 사용자의 요구에 맞는 최적의 서비스를 수행할 수 있어야 한다. 이를 위해선 서비스에 사용하는 리소스들이 유일하게 서술될 수 있어야 하고, 주변 환경과 사용자 요구, 기존 데이터로부터 추론하여 새로운 환경 정보를 생성하고, 생성된 환경 정보를 바탕으로 계획을 수립하여 실행시킬 수 있어야 한다.

또한, 지능형 서비스 로봇은 하나의 사전 정의된 서비스만을 실행하는 것이 아니라 모델로서 정의된 여러 가지 서비스 모델을 필요에 따라 차례로, 혹은 서로 교체해 가며 실행할 수 있어야 한다. 이를 위해서는 서비스 각각이 모델 기반으로 서술되어야 한다.

본 논문에서는 모델 기반 클라우드 서비스 연계 방법의 검증에 위해 수평적 접근과 수직적 접근 방법을 사용하였다. 먼저 표준적이고 범용적으로 외부 서비스 인터페이스를 정의할 수 있는 온톨로지를 작성하여 다양한 클라우드 서비스를 서술할 수 있도록 수평적인 개방성과 확장성을 검증하였으며, 각각의 클라우드 서비스의 실제로 사용 가능한 수준까지의 모델을 작성하고 검증하여 수직적인 구체성을 검증하였다.

4. 클라우드 서비스 모델 및 인터페이스

본 논문에서 제시하는 클라우드 인터페이스는 모델 기반으로 작성되어 모델의 교체 및 추가만으로 연동 가능한 서비스

의 영역을 쉽게 확장할 수 있다는 것이 주된 특징이며, 이를 위하여 클라우드 서비스 활용에 필요한 정보들을 명세화한 클라우드 서비스 프로파일과, 클라우드 서비스 프로파일 내에서 연동 대상이 되는 지능형 로봇 시스템과 클라우드 서비스 간 연계를 위한 클라우드 프로퍼티 모델을 제안한다.

4.1 클라우드 서비스 프로파일

외부 클라우드 서비스는 서비스를 사용하기 위한 다양한 데이터가 필요하다. URL의 주소나 HTTP 메소드와 같은 서비스 자체에 대한 데이터뿐만 아니라 서비스를 사용하는 데 필요한 인증방식 등을 필요로 한다. 이처럼 서비스마다 다른 데이터 요구사항과 인증방식을 포괄적이고 범용적으로 활용할 수 있도록 인터페이스를 작성하기 위해 본 논문에서는 클라우드 서비스 프로파일을 통해 서비스에 필요한 데이터를 정의하고 사용한다.

클라우드 서비스 프로파일을 통해 모델화된 외부 서비스는 클라우드 인터페이스에서 활용되며, 어떤 서비스라도 모델의 교환만으로 쉽게 연계 대상 서비스와 연동할 수 있는 개방성 및 확장성을 확보할 수 있다.

클라우드 프로파일은 두 가지의 구성 요소를 갖는다. 먼저 클라우드 서비스를 활용하는 데 필요한 정보들을 갖는 클라우드 서비스 모델과 클라우드 서비스의 인증 정보를 가진 인증 모델로 구성된다.

Fig. 1은 클라우드 서비스 프로파일의 구성도이다.

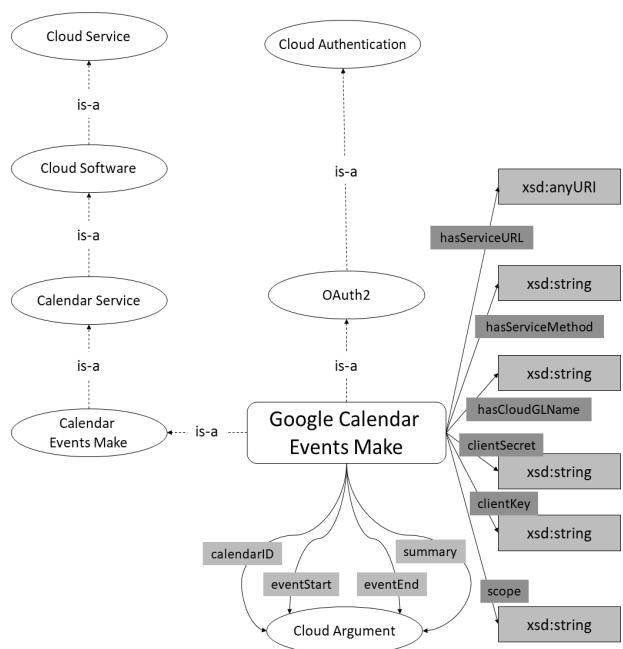


Fig. 1. Cloud Service Profile Architecture

1) 클라우드 서비스 모델

클라우드 서비스 모델은 클라우드 서비스를 활용하는데 필요한 정보들로 구성된다. 이는 서비스에 접근하기 위한 주소

와 HTTP 메소드, 서비스를 활용하는 데 필요한 각종 프로퍼티들을 포함한다(Fig. 3).

서비스 모델에 포함되어야 하는 프로퍼티들은 서비스의 종류마다 상이하기 때문에, Nelson의 Taxonomy Model(Fig. 2) [11]을 참조 및 확장하여 만든 클라우드 서비스 계층 모델을 활용하여 서비스의 특성에 따라 계층적으로 분류한 후, 상위 계층에서 하위 계층으로 갈수록 서비스의 프로퍼티를 구체화해 최종적으로 서비스에 필요한 모든 프로퍼티를 포함할 수 있도록 구성한다.

Table 1은 클라우드 서비스 모델의 구성 요소에 관해 서술한다.



Fig. 2. Nelson's Taxonomy Model

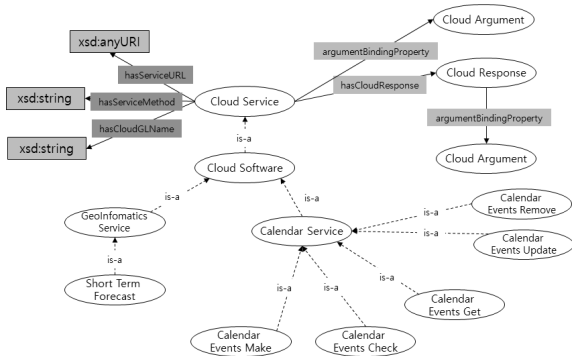


Fig. 3. Cloud Service Model Ontology

Table 1. Cloud Service Model Specification

Name	Description	Type
serviceURL	Cloud Service URL	String
method	Linkage Target Service's HTTP Method	String
protocol	Service Protocol Data Model (will be explained later on)	Protocol Model

2) 인증 모델

클라우드 서비스는 서비스 종류에 따라 개인정보를 요구할 수가 있고, 이로 인해 인증 절차가 필요한 서비스가 존재한다. 이러한 서비스를 사용하기 위해선 인증에 필요한 여러 파라미터로 구성된 인증 모델이 별도로 필요하다.

인증 모델은 현재 구글이나 페이스북, 트위터 등에서 표준적으로 사용 중인 OAuth2 인증방식과 API Key만을 활용하는 인증방식이 존재한다.

OAuth2 인증 모델은 OAuth Authorization 과정에서 필요한 프로퍼티들인 Client Key와 Client Secret, OAuth 인증의 범위를 지정하는 Scope가 필수적으로 포함되며, 개인의 계정 관련 보안 요소인 Refresh Key를 선택적으로 포함할 수 있다.

API Key 인증 모델은 서비스를 사용할 때 필요한 API Key 파라미터만을 요구하는 경우이며, 이 경우에는 API Key만을 포함한다.

Table 2는 OAuth2 기반 서비스의 인증 모델, Table 3은 api key 기반 서비스의 인증 모델이며, Fig. 4는 클라우드 인증 모델의 구성도이다.

Table 2. OAuth2 Authorization Model Specification

Name	Description	Type
provider	Service Provider	String
key	Client Key for authorization process in OAuth2 Cloud API	String
secret	Client Secret for authorization process in OAuth2 Cloud API	String
scope	Service Scope for authorization process in OAuth2 Cloud API	String

Table 3. API Key Authorization Model Specification

Name	Description	Type
key	API Key for api-key based authorization process	String
location	Where api key is assign	Body/Header
bindingName	Binding Target of API Key	

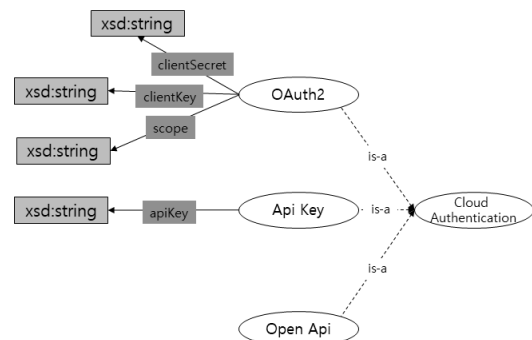


Fig. 4. Cloud Authentication Model Ontology

4.2 클라우드 프로퍼티 모델

클라우드 서비스에 사용되는 클라우드 프로퍼티는 서로 다른 두 시스템을 연계할 필요가 있어 다양한 정보들을 요구한다. 프로퍼티는 클라우드 서비스의 데이터와 맵핑되기 위한 맵핑 룰과 연계 대상 시스템과 연계하기 위한 시스템 맵핑 룰이 필요하고, 또 클라우드 서비스의 종류에 따라 값이 지속적으로 갱신될 필요가 있거나, 일회성으로 요청하거나 조건에 따라 값을 갱신하는 등 프로토콜 형식에 따른 Protocol Type이 필요하다. 또한, 클라우드 서비스 내에서 서술되는 리소스를 정의하고 재활용하기 위해 클라우드 리소스를 정의할 수 있어야 한다.

Fig. 5는 클라우드 프로퍼티 모델의 구성도이다.

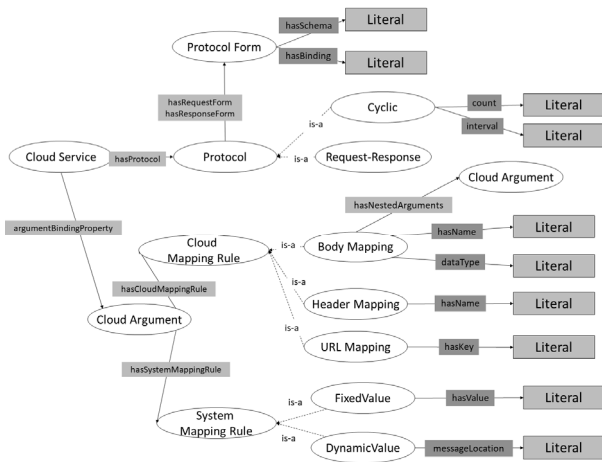


Fig. 5. Protocol Model and Argument Model Ontology

1) 클라우드 맵핑 룰

외부 클라우드 서비스는 웹상의 특정 주소에 HTTP 요청을 수행함으로써 HTTP 응답 데이터를 받아 오는 형식으로 이뤄진다. 이러한 요청 - 응답 과정은 서비스의 URL 주소에 정해진 Header 파라미터와 Body 파라미터를 전달함으로써 실행된다. 맵핑 룰은 클라우드 프로파일에서 지정한 서비스의 파라미터가 HTTP 요청의 구성 요소 중 어느 위치에 배치되어야 하는지를 나타내는 구성 요소이다. 이는 다음과 같이 서술된다.

- a) URL Mapping : URL 주소는 외부 클라우드 서비스의 직접적인 주소를 나타내지만, HTTP의 GET 메소드나 DELETE 메소드와 같이 HTTP 요청 메시지에 Body 항목이 존재하지 않아 주소에 요구 사항을 기재해야 하는 경우가 존재한다. 또한, 특정 리소스를 대상으로 하는 경우 리소스의 주소 자체가 필요에 따라 변경될 필요가 존재한다. 이 두 가지의 요구사항을 충족시키기 위해 본 논문에서는 두 가지 방식의 URL Mapping 방식을 고안하였다. 첫 번째 Mapping 방식은 일반적인 URL Query에 대응하는 방식이다. HTTP의 URL에 Query문을 넣는 방식

은 URL 주소 이후에 특정 문자열을 입력하는 것인데, 이 문자열들은 키-값 쌍으로 구성된다. 따라서 URL에 추가할 키-값 쌍을 바탕으로 리소스 주소 뒤에 추가적인 인자들을 나열하는 방식이다.

두 번째 Mapping 방식은 특정 리소스의 URL을 찾기 위해 서비스의 URL 주소 중 일부 값을 임의로 변경시키는 방식이다. 이를 위해 서비스 프로파일에 서술되는 URL은 변동 가능한 값에는 특수 문자를 사용해 클라우드 인터페이스에서 언제든지 외부 요청에 따른 임의 값으로 주소를 변경시킬 수 있도록 한다.

URL Mapping 방식은 Table 4와 같이 서술된다.

Table 4. URL Mapping Rule Specification

Name	Description	Type
key	Mapping target string	String

- b) Body Mapping : HTTP 요청/응답은 HTTP 메소드에 따라 Body에 해당하는 영역에 인자를 삽입할 필요가 있다. 이러한 인자는 평문 형식부터 JSON, XML 등의 표준적인 Semi-structure 형식의 데이터로 구성될 수 있다. 따라서 Body Mapping을 위한 인자들은 해당 Body문의 형식과 문서 내에 어떤 위치의 인자인지를 서술할 필요가 있다. 본 논문에서는 JSON 형식의 Body Mapping과 XML 형식의 Body Mapping 룰에 대해서만 서술한다.

첫 번째로, XML 형식의 Body Mapping은 파라미터가 맵핑될 대상 위치의 경로를 추가적으로 기재한다. 이를 통해 하나의 파라미터를 XML 문서 내의 특정 위치에 기술할 수 있게 된다.

두 번째로 JSON 형식의 Body Mapping 또한 XML 형식과 마찬가지로 대상 위치의 경로를 Dot notation을 사용해 표기한다. 이를 통해 JSON 객체 내의 특정 위치에 대상 파라미터를 대응시킬 수 있다.

Body Mapping 방식은 Table 5와 같이 서술된다.

Table 5. Body Mapping Rule Specification

Name	Description	Type
name	Mapping Target variable name	String
type	Target variable's resource type	String

- c) Header Mapping : HTTP 요청/응답 프로토콜은 그 외에도 요청/응답에 필요한 메타데이터나 인증과 관련된 추가적인 정보를 담는 Header 영역이 존재한다. HTTP Header는 키-값 쌍으로 구성되기 때문에, Header Mapping은 HTTP Header에 들어갈 키 값의 이름을 서술하게 되어 있다.

Header Mapping 방식은 Table 6과 같이 서술된다.

Table 6. Header Mapping Rule Specification

Name	Description	Type
key	Mapping Target variable name	String

이렇게 서술된 Mapping Rule은 Fig. 6과 같이 서비스 프로파일에서 서술된 내용을 바탕으로 연계 대상 시스템의 메시지와 대응하여 서비스 사용을 위한 HTTP 요청/응답 프로토콜에 사용된다.

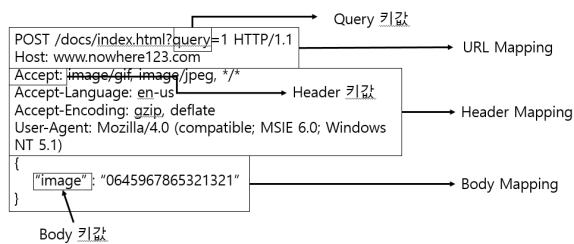


Fig. 6. Mapping Rule Example

2) 시스템 맵핑 룰

클라우드의 파라미터는 적절한 형태로 가공되어 통합 대상이 될 시스템의 데이터로 맵핑될 필요가 있다. 시스템 맵핑 룰은 클라우드의 요청 - 응답 과정에서 해당 데이터가 연계 대상 시스템의 메시지 형식의 어느 부분과 맵핑되는지를 나타내는 데이터이다.

이 때, 맵핑 대상이 되는 데이터는 경우에 따라 서비스 정의 당시 고정된 값이 그대로 사용될 수 있어야 한다. 따라서 시스템 맵핑 룰 프로토콜은 서술 대상이 고정값인지 유동값인지 확인하기 위한 서술이 필요하다.

시스템 맵핑 룰은 Table 7과 같이 서술된다.

Table 7. System Mapping Rule Specification

Name	Description	Type
isFixed	Boolean that express value is fixed value or dynamic value.	boolean
name	Mapping target location	String
value	Mapping target value	String

3) 프로토콜 타입

클라우드 서비스는 활용 방법에 따라 한 번의 요청에 한 번의 실행, 혹은 지속적인 갱신을 요구할 수 있다. 또, 클라우드 서비스가 실행된 뒤 응답받은 서비스 데이터가 연계 대상 시스템 메시지 형식과 일치시킬 필요가 있다.

프로토콜 타입은 이러한 여러 형태의 서비스 접근 방식을 분류하고 실행시킨 후, 응답받은 데이터를 연계 대상의 메시지와 연계시키기 위해 존재한다.

프로토콜 타입은 Request-Response 형식과 Cyclic 형식이 존재한다.

a) Request-Response 형식 : 일반적인 HTTP 요청-응답 방식과 동일하게 한 번에 하나의 요청/응답만을 처리하는 프로토콜이다. Request-Response 형식의 프로토콜 타입은 연계 대상의 메시지와 대응시키기 위한 데이터만을 포함한다.

Table 8은 Request-Response Data Model에 관해 서술한다.

Table 8. Request-Response Data Spec

Name	Description	Type
request	Request schema data	(depend on system)
request Binding	A pair of Keyword that bind into request schema data.	Key/Value Pair
response	Response schema data	(depend on system)
response Binding	A pair of Keyword that bind into response schema data.	Key/Value Pair

b) Cyclic 형식 : 일정 주기마다 반복적으로 HTTP 요청-응답을 반복해야 하는 프로토콜이다. Request-Response 프로토콜 타입의 데이터에 추가로 프로토콜 반복 주기를 설정하는 데이터를 포함한다.

Table 9는 Cyclic Data Model에 관해 서술한다.

Table 9. Cyclic Data Spec

Name	Description	Type
request	Request schema data	(depend on system)
request Binding	A pair of Keyword that bind into request schema data.	Key/Value Pair
response	Response schema data	(depend on system)
response Binding	A pair of Keyword that bind into response schema data.	Key/Value Pair
interval	Interval of request/response	int(ms)
count	repeat count	signed int (0 on infinity)

4) 클라우드 리소스

클라우드 서비스의 요청/응답에 활용되는 데이터들은 특정 형식을 갖고 있으며, 경우에 따라선 데이터가 다른 형식의 데이터를 포함하거나, 같은 종류의 데이터를 여러 갖고 있는

등 특정한 형식의 데이터를 반복적으로 사용해야 할 필요가 있다.

이를 위해 클라우드 인터페이스 모델에서는 클라우드 서비스에 사용되는 각 리소스를 정의 및 재사용할 수 있다.

본 논문에서는 클라우드 리소스의 정의 및 활용을 위하여 JSON Schema[12]를 사용하였다.

5. 실험

본 논문에서는 지능형 로봇의 서비스 영역을 확대하는 방안으로 클라우드 서비스의 모델 기반 연계 방법을 제안하였다. 이를 검증하기 위해서 다른 인증방식을 가진 2종의 클라우드 서비스를 대상으로 클라우드 서비스 모델을 작성하여 클라우드 서비스 모델이 서술하고자 하는 서비스의 인증방식이나 Argument 전달 방식 등과 관계없이 서술할 수 있고, 하나의 서비스를 대상으로 본 논문에서 제안한 인터페이스를 통해 클라우드 서비스를 사용하는 실험을 수행하여 제안된 인터페이스가 로봇 시스템과 연계되는 경우에 서비스 로봇 프로토콜과 클라우드 서비스 요청 간 변환이 가능한지에 대해 검증하였다.

5.1 클라우드 서비스 모델 검증

이 실험에서는 접속 인증 방법 및 서비스의 종류에 편차가 있는 2종류의 서비스를 대상으로 클라우드 서비스 모델 스키마가 구성 가능한지를 검증하였다. 실험 대상 서비스는 Google Calendar API와 기상청 예보문 API 서비스이다.

구글 캘린더 API는 OAuth2 인증방식을 사용한다. 따라서 OAuth2 기반 인증 모델을 사용해야 하며, 인증 과정에서 사용자에게 리소스 사용 권한을 취득해야 하고, 요청 과정에서 인자값은 URL 주소에 Query 방식으로 추가되는 것이 아닌 특정 영역의 키워드를 교체하는 방식으로 이루어진다.

기상청 예보문 서비스는 API Key 인증방식을 사용하여 별도의 추가 인증 과정 없이 서비스를 활용할 수 있고, 서비스에 사용되는 위도, 경도나 시각 등은 정해진 리소스 URL 주소에 URL Query 방식으로 추가된다.

Table 10은 각 클라우드 서비스의 모델 상세를 나타낸 것이다. 두 개의 클라우드 서비스는 서로 다른 인증방식을 갖고 있으며, 이는 클라우드 서비스 모델 스키마의 OAuth2 인증 모델과 API key 인증 모델을 통해 서술되었다.

또한, 서비스의 인자값을 할당하는 방식도 각각 URL 내의 키워드를 치환하는 방법과 URL Query 양식을 사용하는 방법으로 상이하지만, URL 맵핑 룰에 의해 두 방식 모두를 서술할 수 있었다.

이 중 구글 캘린더는 기상청 API와 다르게 Response 값 내에 반복적으로 서술되는 데이터가 존재하기 때문에 클라우드 리소스 모델을 통해 클라우드 서비스 프로토콜에서 다른 프로토콜로의 변환 방법과 맵핑 룰을 서술하였다.

이러한 모델 작성 실험을 통해 OAuth2와 API key 방식의 서로 다른 인증방식과 서비스 내용이 상이한 2종의 클라우드 서비스가 모두 클라우드 서비스 스키마에 의해 서술 가능함을 확인하였다.

5.2 모델 기반 클라우드 서비스 연계 검증

본 실험에서는 비서 기능을 수행하는 서비스 로봇이 일정에 따라 안내를 수행하거나, 신규 일정에 대한 시간적 충돌 등을 추론하여 서비스를 수행하기 위하여 클라우드상에 있는 일정 정보를 획득하는 상황을 가정하였다.

이때, 인터페이스는 실험 5.1.을 통해 작성된 Google Calendar API의 클라우드 모델을 활용하여 서비스 로봇의 요청으로 일정 확인 서비스를 실행하여 로봇에게 적합한 형태로 가공된 사용자의 일정 정보를 전달해야 한다.

실험 과정은 다음과 같이 진행된다. 1) Google Calendar API의 클라우드 서비스 모델을 모델 기반 인터페이스에 적용한다. 2) 서비스 로봇이 Google Calendar 사용 요청을 인터페이스에게 전달한다. 3) 인터페이스가 서비스 로봇의 요청과 클라우드 서비스 모델을 통해 클라우드 서비스 인증 과정 및 요청/응답을 수행한다. 4) 인터페이스가 클라우드 서비스로부터 반환받은 값을 모델을 통해 변환하여 서비스 로봇에게 전달한다.

위와 같은 과정을 통해 실험을 진행한 결과, Fig. 7과 같이 구글 캘린더로부터 서비스 로봇이 요청한 일정 정보를 가져 오고, 일정 정보를 로봇이 필요한 형태로 가공하여 전달하는 과정을 확인할 수 있었다.

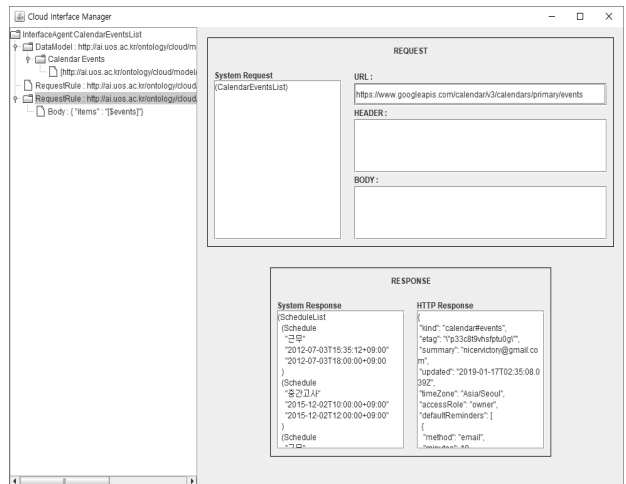


Fig. 7. Result of Experiment 5.2

본 실험을 통해 클라우드 모델에 작성된 파라미터나 인증 방식을 통해 서비스가 실행되는 것을 확인하였으며, 나아가 클라우드 맵핑 룰을 통해 서비스 로봇 프로토콜과 클라우드 서비스 프로토콜 간 변환을 검증하였다.

Table 10. Google Calendar List Request-Response Data Spec

Name		Google Calendar	KMA Forecast
Resource	Event	{ "type": "object", "properties": { "calendarID": { "type": "string" }, "start": { "type": "object", "properties": { "dateTime": "#start" } }, "end": { "type": "object", "properties": { "dateTime": "#end" } }, "summary": { "type": "string" } } }	N/A
Request	URL Schema	GET: https://www.googleapis.com/calendar/v3/calendars/\$calendarID/events	GET: http://newsky2.kma.go.kr/service/VilageFrstDspthDocInfoService/WidGeneralWeatherCondition
	Cloud Argument	URL Mapping : key : \$calendarID, System Mapping : name : calendarID, isFixed : false	URL Mapping : key : stnId Body Mapping : name : stationId, isFixed : false
	Authorization	OAuth2 { "provider" : "google", "api key" : "*api key*", "api secret" : "*api secret*", "scope" : "https://www.googleapis.com/auth/calendar.events.readonly" }	APIKey("*api key*")
	Robot Message Example	QUERY:(CalendarEventsList calendarID)	QUERY:(WeatherAnnouncement stationId)
Response	Body Schema	{ "type" : "object", "properties" : { "items" : { "type" : "array", "items" : Event } } }	{ "type" : "object", "properties" : { "response" : { "type" : "object", "properties" : { "body" : { "type" : "object", "properties" : { "items" : { "type" : "object", "properties" : { "item" : { "type" : "object", "properties" : { "wfSv1" : "string" } } } } } } } } } } }
	Cloud Argument	"Body Mapping" : { name : items, type : Event }, "System Mapping" : { name : eventList }	{ "Body Mapping" : { name : response.body.items.item.wfSv1, type : string }, System Mapping : { name : WeatherAnnouncement, isFixed : false } }
	Robot Message Example	RESPONSE : (ScheduleList (eventList (event #start #end)))	RESPONSE : (WeatherAnnouncement announcement)

6. 결 론

본 논문에서는 지능형 서비스 로봇을 위한 모델 기반 클라우드 서비스 인터페이스를 제안하였다. 지능형 서비스 로봇과 같이 범용적인 서비스를 수행하는 환경에서는 능동적으로 서비스를 수행할 방법이 제공되어야 하며, 이러한 인터페이스는 모델을 통해 외부 서비스를 연동하는 방법을 제공할 수 있어야 한다. 이때, 모델은 클라우드 서비스를 범용적으로 서술할 수 있는 명세에 기반하여 작성되어야 하며, 서비스에 대한 명세는 서비스 정보 및 인증 방법을 서술할 수 있어야 하고, 이를 통한 자동적인 데이터의 변환 및 맵핑이 가능하여야 한다.

본 논문에서 제안한 클라우드 서비스 모델은 클라우드 환경에서 사용 가능한 소프트웨어 서비스에 대해 지식 모델을 구축하여 개방성 및 확장성을 제공하였고, 이 모델은 서비스의 구성과 인증 방법을 축으로 하는 2차원 모델로 구성하여 자동으로 이를 수행할 수 있도록 하였으며, 이를 위한 자동적인 인터페이스 모듈을 생성하는 방법을 통해 기반 시스템이 쉽게 서비스를 활용할 수 있는 방법을 개발했다.

또한, 본 논문에서 제안한 지능형 서비스 로봇을 위한 모델 기반 클라우드 서비스 인터페이스는 실험을 통해 다수의 서비스에 대해 검증을 수행했고, 향후 다른 서비스에 대한 확장성 또한 검증할 수 있었다.

그러나 현재의 서비스는 RESTful API나 구글 OAuth 2 Credential과 같이 제한적인 접속 인증방식에 한정되어 구성되어 있으며, 또한 기반 시스템으로 활용하는 대상을 로봇에 대해 한정하고 있어 데이터에 대한 범용적인 맵핑 방법을 제공하지 못했다.

따라서 향후 연구에서는 더욱 다양한 서비스에 대한 모델링을 통해 인터페이스를 통해 제공할 수 있는 다른 인증방식 등의 개방성을 추구할 것이며, 또한 기반 시스템으로 활용할 수 있는 다양한 시스템 및 통신방법을 지원할 수 있는 범용적 맵핑 모델을 구축하여야 할 것이다.

References

[1] U. Kurup and C. Lebiere, "What can cognitive architectures do for robotics?" *Biologically Inspired Cognitive Architectures*, Vol.2, pp.88-99, 2012.

[2] C. M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, R. Metz, and B. A. Hamilton, "Reference Model for Service Oriented Architecture 1.0", OASIS Standard, 12 October 2006, <http://docs.oasis-open.org/soa-rm/v1.0>.

[3] R. France and B. Rumpe, "Model-Driven Development of Complex Software: A Research Roadmap," *Future of Software Engineering*, L. Briand and A. Wolf, eds., IEEE CS Press, 2007.

[4] RAML [Internet], <https://raml.org>

[5] Niklas Scholz, Adrian Hernandez-Mendez, Prof. Dr. Florian Matthes, "A Model-Based Approach to Consume REST Services in Single Page Applications," Department of informatics, TECHNISCHE UNIVERSITÄT MÜNCHEN, Master's thesis in informatics

[6] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana, "Unraveling the Web Services Web : An Introduction to SOAP, WSDL, and UDDI," in *IEEE Internet Computing*, Vol 6, Issue 2, IEEE, 2002.

[7] Oren Ben-Kiki, Clark Evans, Ingy döt Net, *YAML Ain't Markup Language (YAML™ Version 1.,2) 3rd edition*, [Internet], <https://yaml.org/spec/1.2/spec.html>,

[8] SOAP Version 1.2 Part 0 : Primer, [Internet], <https://www.w3.org/TR/2007/REC-soap12-part0-20070427>, W3C.

[9] UDDI Version 3.0.2, UDDI Spec Technical Committee Draft, Dated 20041019, [Internet], <https://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm>, W3C.

[10] Web Service Description Language(WSDL) Version 2.0 Part 0 : Primer, [Internet], <https://www.w3.org/TR/wsdl20-primer>, W3C.

[11] N. M. Gonzalez, C. Miers, F. F. Redigolo, M. A. Simplicio Jr, T. C. M. Carvalho, M. Näslund, and M. Pourzandi, "A Taxonomy Model for Cloud Computing Services," In *Proceedings of the 1st International Conference on Cloud Computing and Services Science*, pp.56-65, 2011.

[12] JSON Schema [Internet], <https://datatracker.ietf.org/doc/draft-handrews-json-schema>, JSON Schema.



최 병 기

<https://orcid.org/0000-0003-4202-2489>

e-mail : byunggi.choi@gmail.com

2010년 서울시립대학교

전자전기컴퓨터공학부(학사)

2012년 서울시립대학교

전자전기컴퓨터공학과(석사)

2012년 ~ 현재 서울시립대학교 전자전기컴퓨터공학과 박사과정
관심분야 : 인공지능, 소프트웨어 프레임워크, 작업관리



이 종 욱

<https://orcid.org/0000-0001-7890-0197>
e-mail : uosbell@gmail.com
2016년 ~ 현 재 서울시립대학교
정보기술연구소 연구원
관심분야 : 웹 기술, 소프트웨어 프레임워크,
인공지능



이 재 호

<https://orcid.org/0000-0002-3332-3207>
e-mail : jaeho@uos.ac.kr
1985년 서울대학교 계산통계학과(학사)
1987년 서울대학교 계산통계학과(석사)
1997년 University of Michigan(박사)
1998년 ~ 현 재 서울시립대학교
전자전기컴퓨터공학부 교수

관심분야 : 인공지능, 지능 로봇



박 성 기

<https://orcid.org/0000-0002-4278-7206>
e-mail : skee@kist.re.kr
1987년 서울대학교 기계설계학과(학사)
1989년 서울대학교 기계설계학과(석사)
2000년 한국과학기술원 자동화설계(박사)
2000년 ~ 현 재 한국과학기술연구원
지능로봇연구단 책임연구원

관심분야 : 로봇지능, 인공지능, 컴퓨터비전