

# Dynamic Service Composition and Development Using Heterogeneous IoT Systems

Minwoo Ryu\*, Jaeseok Yun\*\*

## Abstract

IoT (Internet of Things) systems are based on heterogeneous hardware systems of different types of devices interconnected each other, ranging from miniaturized and low-power wireless sensor node to cloud servers. These IoT systems composed of heterogeneous hardware utilize data sets collected from a particular set of sensors or control designated actuators when needed using open APIs created through abstraction of devices' resources associated to service applications. However, previously existing IoT services have been usually developed based on vertical platforms, whose sharing and exchange of data is limited within each industry domain, for example, healthcare. Such problem is called 'data silo', and considered one of crucial issues to be solved for the success of establishing IoT ecosystems. Also, IoT services may need to dynamically organize their services according to the change of status of connected devices due to their mobility and dynamic network connectivity. We propose a way of dynamically composing IoT services under the concept of WoT (Web of Things) where heterogeneous devices across different industries are fully integrated into the Web. Our approach allows developers to create IoT services or mash them up in an efficient way using Web objects registered into multiple standardized horizontal IoT platforms where their resources are discoverable and accessible. A Web-based service composition tool is developed to evaluate the practical feasibility of our approach under real-world service development.

▶Keyword: Internet of Things, Dynamic service composition, Standards, Web of Things, Semantic description, Service description, Interoperability

## I. Introduction

사물인터넷 (IoT: Internet of Things)은 실세계에 존재하는 물리사물 (Physical Things)과 사이버 세계에 존재하는 가상사물 (Virtual Things)들이 Bluetooth, ZigBee, WiFi 등의 네트워크 기술을 통하여 연결되고 이를 기반으로 다양한 서비스를 제공할 수 있는 기술을 말한다 [1]. 따라서, IoT 시스템은 초소형, 저전력 무선 센서 노드부터 대형 클라우드 서버에 이르기까지 다양한 종류의 시스템들로 구성되며 서로 다른 네트워크를

통해 상호 연결되어 정보 공유가 가능하게 되었다. 이를 통해 수집된 데이터는 분석작업을 통해 특정 상황을 인식하거나, 이에 대응하여 물리 사물을 제어하는 등의 서비스를 사용자에게 제공할 수 있다. 이러한 장점으로 인하여 IoT 시스템은 최근 헬스케어, 스마트 시티, 스마트 팜 등 다양한 분야에서 활용되고 있다 [2-6].

현재 IoT 시스템을 활용하여 개발되고 있는 서비스 애플리

• First Author: Minwoo Ryu, Corresponding Author: Jaeseok Yun

\*Minwoo Ryu (mw.ryu@kt.com), Service Laboratory, Institute of Convergence Technology, KT R&D Center

\*\*Jaeseok Yun (yun@sch.ac.kr), Dept of Internet of Things, SCH Media Labs, Soonchunhyang University

• Received: 2017. 08. 01, Revised: 2017. 08. 11, Accepted: 2017. 08. 24.

• This work was supported by the Soonchunhyang University Research Fund (No.20170693).

• This work was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2017R1D1A3B03032676).

케이션들은 물리사물과 연결된 가상사물의 리소스를 추상화하고, 이를 통하여 생성된 범용 인터페이스 (예: RESTful API)를 활용하고 있다 [7]. 하지만, 이와 같이 개발된 IoT 서비스들은 특정 도메인에 종속되거나 또는 연동된 플랫폼의 특성에 따라 수직 구조적으로 개발되고 있기 때문에 서비스 간의 결합 또는 연동 시 기존 개발된 서비스를 편집하거나 또는 재활용하기 위해서는 많은 제약사항을 해결해야 한다. 예를 들어, 스마트홈과 헬스케어 분야에서 사용되는 디바이스들은 활용 분야와 사용자 정보가 밀접히 관련이 있어, 서로간 정보 공유와 협업이 긴밀히 이루어질 수 있으나, 현재 대부분 관련 제품들은 각자 자신만의 (proprietary) 수직적 IoT 플랫폼을 기반으로 개발되기 때문에 현실적으로 협업 서비스 구성이 어렵다. 뿐만 아니라, IoT 서비스는 네트워크나 모바일 기기의 이동 특성에 따라 연결된 기기들의 상태가 수시로 변화하기 때문에 이러한 변화에 대응하는 동적 IoT 서비스 구성 방법이 필요하다.

이를 위하여, 본 논문에서는 다양한 서비스 도메인에 등록된 이종의 IoT 디바이스를 활용하여 동적으로 서비스를 구성할 수 있는 방법을 제안한다. 제안하는 동적 서비스 구성 방법은 IoT 디바이스의 프로파일 (예, 기능, 종류 등)을 시맨틱 기술을 이용하여 웹 오브젝트로 생성하여, 개발자들이 서비스 구성 시 쉽고 편리하게 IoT 디바이스 리소스에 접근할 수 있도록 한다. 또한, 서비스 컴포지션 기술을 통하여 생성된 웹 오브젝트들 간의 결합을 제공함으로써, 단일 서비스뿐만 아니라 복합 서비스를 개발할 수 있는 환경을 제공한다. 마지막으로 웹 기반 편집 툴 개발을 통해 제안된 방법론의 실제 시스템 개발에서의 효율성을 검증한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안하는 동적 서비스 구성 방법에 대한 기존 관련연구에 대하여 살펴보고, 3장에서는 웹오브젝트를 활용한 동적 서비스 구성 방법에 대하여 논한다. 그리고 4장에서는 제안하는 동적 서비스 구성 방법의 효율성을 입증하기 위하여, 웹 기반 편집 툴을 통하여 실제 시스템 개발에서의 효율성을 검증한다. 마지막으로 5장에서는 결론과 향후 과제를 논한다.

## II. Related Works

본 절에서는 기존 연구되었던 서비스 컴포지션 기술에 대하여 살펴본다. 서비스 컴포지션은 서비스 중심 아키텍처 (SOA, Service Oriented Architecture) [8]의 개념에서 시작한 것으로서, 서비스 단위의 결합을 의미한다.

Brønsted는 퍼베이시브 컴퓨팅에서 기존 서비스들을 결합하여 신규 서비스를 생성할 수 있는 서비스 컴포지션 매커니즘에 대한 기술을 소개하였다 [9]. 해당 논문에서는, 약 20여개 이상의 논문에서 제안되었던 다양한 서비스 컴포지션 매커니즘을 컴포지션 주체 (사용자, 애플리케이션 개발자), 시간 (실시

간, 개발과정), 방법 (사용자 인터랙션, 설정, 소스코드), 과정 (명시적, 내포적), 서비스 수준 요구사항 유무 등을 분석하였다. 또한 해당 서비스 컴포지션 방법을 실제로 구현하기 위해 필요한 기기와 인프라 등 요구사항들을 조사하였다.

Kalasapur는 퍼베이시브 컴퓨팅 환경에서 동적으로 서비스를 결합하기 위한 서비스 컴포지션 매커니즘 모델을 제안하였다 [10]. 제안하는 모델은 그래프 이론을 기반으로 서비스 컴포지션 매커니즘을 제공하여, 퍼베이시브 컴퓨팅 환경에서 계층적인 서비스 구조를 제안하였다. 뿐만 아니라 서비스 컴포지션 미들웨어인 PICO (Pervasive Information Communities Organization) 플랫폼을 이용하여 서비스를 구성하고 서비스 요구사항을 관리할 수 있는 체계를 제공하였다 [11]. 하지만 제안하는 서비스 모델은 기존 구성된 서비스들을 결합하거나 조합하는 장점은 가지고 있지만, 동적으로 증가되는 IoT 시스템 또는 디바이스를 수용하기에는 많은 제약 사항이 따른다.

Sheng은 웹 기반의 서비스 컴포지션 기술에 대하여 소개하였다 [12]. 서비스 컴포지션에 대하여 표준, 프로토타입, 플랫폼과 관련된 기술 동향을 조사하였다. 특히 서비스 컴포지션에 대한 전반적인 라이프 사이클과 서비스 컴포지션에 대한 표준화 기술, 플랫폼에 대한 핵심 기술을 소개하였다. 뿐만 아니라, 서비스 컴포지션에 대한 주요 연구 과제와 해결점에 대한 방향을 제시하였다.

앞서 소개한 서비스 컴포지션에 대한 기존 기술은 서비스 중심 아키텍처에서 이미 개발된 서비스를 결합하는데 초점이 맞춰져 있다. 하지만 IoT 시스템에서는 리소스 중심의 아키텍처 (ROA, Resource Oriented Architecture) [13] 를 중심으로 개발되고 있기 때문에 ROA와 SOA를 연결할 수 있는 서비스 컴포지션 방법이 필요하다. 이를 위하여 본 논문에서는 IoT 플랫폼에 등록된 리소스를 SOA 구조에 맞춰 시맨틱 기술을 기반으로 웹 오브젝트화 하고, 이를 통하여 동적으로 서비스를 구성할 수 있는 방법과, 이를 통하여 제공하는 웹 기반 편집툴을 제안한다.

## III. Dynamic Service Composition

### 1. System overview

본 절에서는 제안하는 동적 서비스 구성방법에 대한 전체적인 개념에 대하여 기술한다. 제안하는 동적 서비스 구성 방법은 특정 도메인에 맞춰 제한된 서비스를 구성하는 것이 아닌 IoT 디바이스의 프로파일을 웹 오브젝트로 생성하여, 도메인 구분 없이 개발자가 쉽고 편리하게 서비스를 구성할 수 있게 한다. 또한, 생성된 웹 오브젝트들의 결합을 통하여 단일 서비스를 구성하는 것 뿐만 아니라 복합 서비스를 구성할 수 있는 환경을 제공한다. 그림 1은 제안하는 동적 서비스 구성 방법에 대한 개념을 도식화 한 것이다.

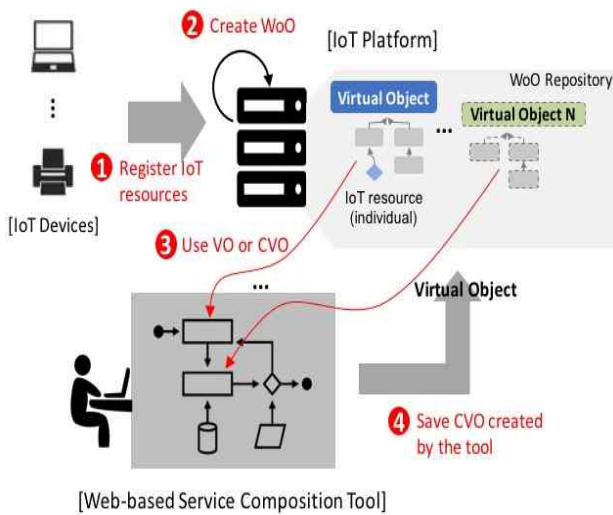


Fig. 1. Our system overview

먼저, 그림 1-①과 같이 IoT 디바이스는 IoT 플랫폼에 등록된다. 이때 등록되는 정보는 디바이스의 종류, 기능 타입 (조회, 제어), 명령어 등이 포함된다. IoT 디바이스의 플랫폼 등록을 완료하게 되면, IoT 플랫폼에서는 등록된 정보를 기반으로 그림 1-②와 같이 하나의 단일 가상 오브젝트 (VO, Virtual Object)로 생성된다. 이후, 개발자들은 원하는 서비스를 쉽게 개발하거나 기존 서비스를 효율적으로 재구성하기 위해 IoT 플랫폼에 등록된 VO를 활용하게 된다. 이를 위하여 본 논문에서는 웹 기반 서비스 편집 툴을 제공한다. 웹 기반 편집 툴은 별도의 프로그램 설치 없이 웹 브라우저를 통하여 서비스를 구성할 수 있는 환경을 제공한다. 따라서 서비스 구성 시 IoT 플랫폼에 등록된 VO를 이용하여 동적으로 서비스를 구성할 수 있게 한다 (그림 1-③). 마지막으로 구성이 완료된 서비스는 그림 1-④와 같이 IoT 플랫폼에 저장된다. 이때 저장되는 오브젝트는 VO가 결합된 복합 가상 오브젝트 (CVO, Composite Virtual Object)로서, 다수의 IoT 디바이스의 결합 (즉, 각 디바이스가 수집하는 데이터, 또는 디바이스를 구동하기 위한 구동 변수를 활용한 매쉬업 과정)을 통하여 복합 서비스를 가능하게 한다.

개발자는 위 과정을 통해 자신만의 CVO를 생성하여 IoT 플랫폼에 저장할 수 있으며, 저장된 CVO를 통하여 실제 환경에 있는 IoT 디바이스를 제어하는 서비스를 실현할 수 있게 한다. 뿐만 아니라, 저장된 CVO는 초기에 구성된 서비스를 편집 및 변경하거나, 서비스 간 연동을 통합 복합 서비스를 쉽게 구성할 수 있을 뿐만 아니라, 이동형 디바이스나 네트워크 환경 변화로 인한 주변 환경 변화에 따라 동적으로 대응하여 서비스 구성 및 실행이 가능하게 된다.

## 2. Web Object-based Structure

본 절에서는 서비스 구성 시 사용되는 VO와 CVO의 구조에 대하여 기술한다.

### 2.1 Virtual Object

VO는 IoT 디바이스를 웹 오브젝트로 표현한 것으로서 IoT 디바이스의 프로파일에 대한 정보를 정의한다. 이를 위하여 본 논문에서는 시맨틱 기술을 이용하여 IoT 디바이스의 프로파일에 대한 의미를 주석화하고, 서비스 구성 시 해당 의미를 추출하여 개발자가 IoT 디바이스의 지식이 없이도 서비스를 쉽게 구성할 수 있게 한다. 이를 위하여 우리는 IoT 국제 표준 협의체인 oneM2M [14]에서 제시하는 oneM2M Base Ontology [15]의 구조를 따르며, 다양한 이종의 IoT 디바이스의 분류를 정의하기 위하여 센서, 액추에이터 등 다양한 디바이스의 분류 체계를 정의한 M3-Lite Ontology (Machine-to-Machine Measurement Lite Ontology) [16]를 사용하였다. 그림 2는 VO의 구조를 보인 것이다.

그림 2에서, 검은색 동그라미는 클래스를 나타내며, 검은색 화살표는 오브젝트 프로퍼티를 표현한 것이다. 그리고 파란색 화살표는 서브클래스의 관계를 표현한 것이다.

먼저, Thing 클래스는 대상 IoT 디바이스의 최상위 표현으로, Device클래스를 하위 클래스로 가진다. 따라서 IoT 기기를 식별하는 고유의 이름이 하나의 개체 (Individual)로 정의된다. 이때, IoT 디바이스의 종류는 M3-lite에서 정의한 분류체계를 따른다.

Service 클래스는 IoT 디바이스가 제공할 수 있는 서비스를 표현한 클래스이다. 예를 들면, 온도 제어 서비스, 온도 조회 서비스 등의 서비스 이름이 하나의 개체로 정의된다.

Functionality 클래스는 해당 IoT 디바이스의 특정 서비스를 제공하기 위한 기능을 정의한 클래스이며, 기능 구분에 따라 ControllingFunctionality, MeasuringFunctionality 클래스를 하위 클래스로 가진다.

Operation 클래스는 Service 클래스에서 명시한 서비스들의 실행 방법에 대한 것을 표현한 클래스로서, HTTP 프로토콜을 이용하여 IoT 플랫폼에 등록된 IoT 디바이스를 제어하기 위한 방법을 정의한다. 따라서 Operation 클래스는 해당 리소스에 접근하기 위한 GET\_InputDataPoint 클래스와 SET\_OutputDataPoint 클래스를 하위 클래스로 가진다.

Command 클래스는 IoT 디바이스의 고유 기능을 실행하기 위한 명령어를 표현하는 클래스로서, 해당 기기의 기능을 동작시키기 위한 명령어가 하나의 개체로 정의된다. 따라서 Command는 해당 명령을 전달하기 위한 IoT 디바이스 리소스의 URL을 포함하는 OperationOutput 클래스와 OperationInput 클래스와 의미적 연결구조를 가진다.

마지막으로 Variable 클래스는 IoT 디바이스를 표현하기 위하여 필요한 다양하게 정의되는 값을 표현한 클래스로서 Aspect 클래스, Unit클래스 등을 하위 클래스로 가진다.

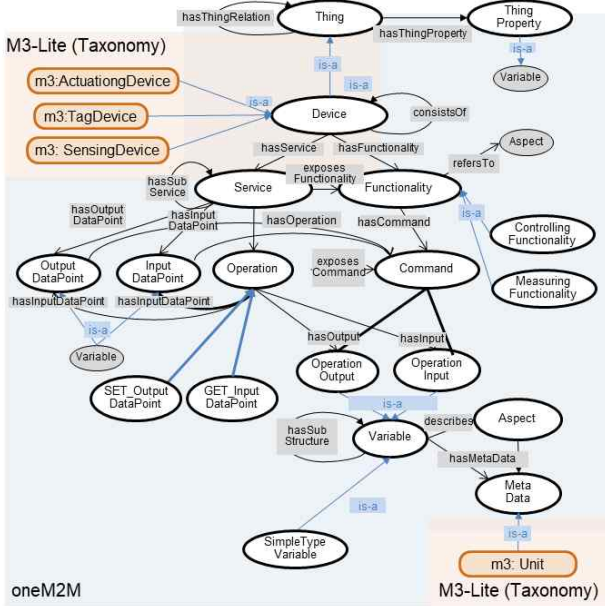


Fig. 2. Semantic-based Web of Object

2.2 Composite Virtual Object

CVO는 VO를 통하여 만들어지는 복합 가상 오브젝트로서, 단일 IoT 디바이스들 간의 연결을 통하여 만들어지는 복합 서비스를 표현하기 위한 오브젝트이다. 그림 3은 CVO의 구조를 보인 것이다. 그림 3에서 검은색 동그라미는 클래스를 표현한 것이며, 검은색 화살표는 오브젝트 프로퍼티를 표현한 것이다. 그리고 초록색 동그라미는 Data를 포함하는 Literal 타입의 클래스를 표현한 것이며, 초록색 화살표는 데이터 프로퍼티를 표현한 것이다. 마지막으로 파란색 화살표는 서브클래스 관계를 표현한 것이다. CVO는 VO들이 결합되어 복합 서비스를 생성할 때 사용되며, 개발자가 웹 기반 편집 툴을 이용하여 동적 서비스를 구성할 때 최종 결과물로 나타난다. 또한 VO를 통한 동적 서비스

구성 이외에, 기존 개발된 서비스를 편집하거나 확장할 때도 해당 CVO를 이용하여 가능하게 한다. 이를 위하여 CVO는 ServiceModel 클래스, Resource 클래스, ServiceProfile 클래스, ServiceGrounding 클래스의 상위 클래스로 나뉜다.

먼저 ServiceModel 클래스는 서비스를 수행하기 위한 단계에 대하여 표현한 클래스로서, 서비스 단계별 사용된 측정, 제어, 조건 확인 등에 대한 서비스 프로세스를 주요 내용으로 포함한다. 따라서 해당 클래스는 개별 프로세스에 필요한 IoT 디바이스와의 의미적 연결을 가진다.

Resource 클래스는 VO에 대한 정보를 포함하는 클래스이다. 따라서 VO에서 사용되는 Device 클래스, Functionality 클래스를 참조한다.

ServiceProfile 클래스는 구성하고자 하는 서비스에 대한 설명을 표현하는 클래스로서, 서비스 분류, 서비스 소유자 등의 정보를 포함한다.

마지막으로 ServiceGrounding 클래스는 서비스 제공 시 해당 IoT 디바이스에 접근하는 방법에 대하여 표현한 클래스이다. 따라서, 개별 단계에서 정의된 VO에 대한 각 정보를 표현하는 URI 정보를 포함하고 있다.

IV. Implementation

본 절에서는 앞서 기술한 동적 서비스 구성 방법을 기반으로 웹 기반 편집 툴 개발에 대하여 기술한다. 개발하는 웹 기반 편집 툴은 별도의 프로그램 설치 없이 인터넷 사용을 위한 웹 브라우저를 통하여 제공되며, IoT 플랫폼 위에서 동작한다. 이를 위하여 우리는 전자부품연구원 (KETI)에서 개발한 모비우스 플랫폼을 이용하였다 [17].

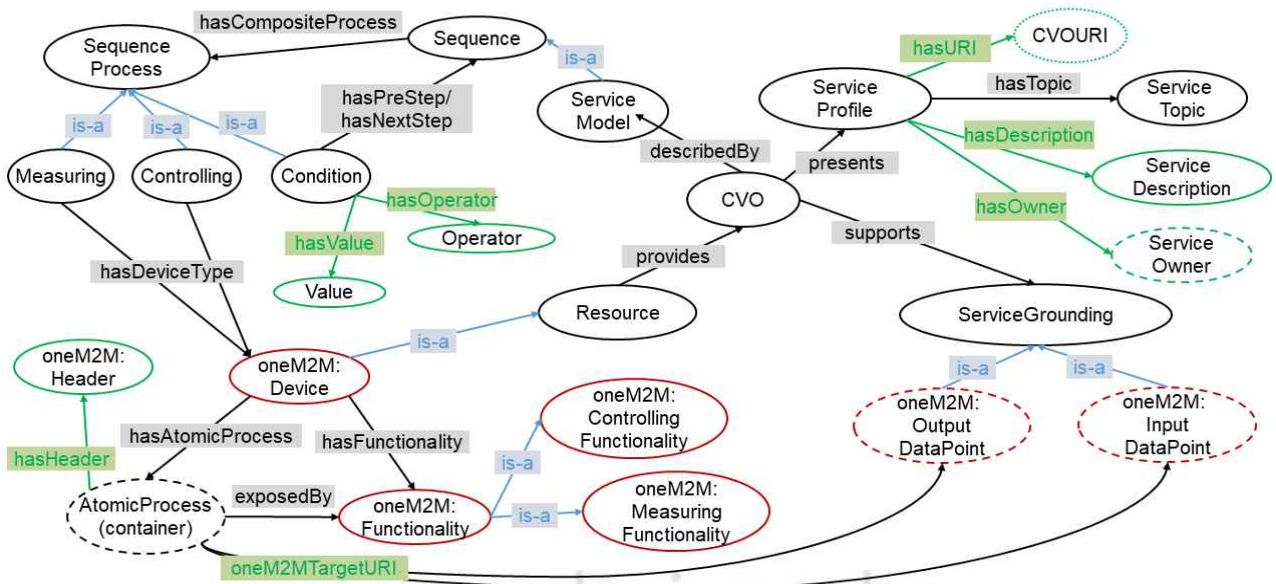


Fig. 3. The structure of composite virtual objects (CVOs)

### 1. VO Implemetation

VO 구현을 위하여, 온톨로지 개발 툴인 프로티지 (protégé) [18]를 이용하였으며, IoT 기기를 가상화하고, 서비스 결합 시 각 디바이스의 기능을 시스템이 이해할 수 있도록 개발 되었다. 그림 4는 프로티지를 이용하여 개발된 VO의 클래스, 오브젝트 프로퍼티, 데이터 프로퍼티를 보인 것이다.

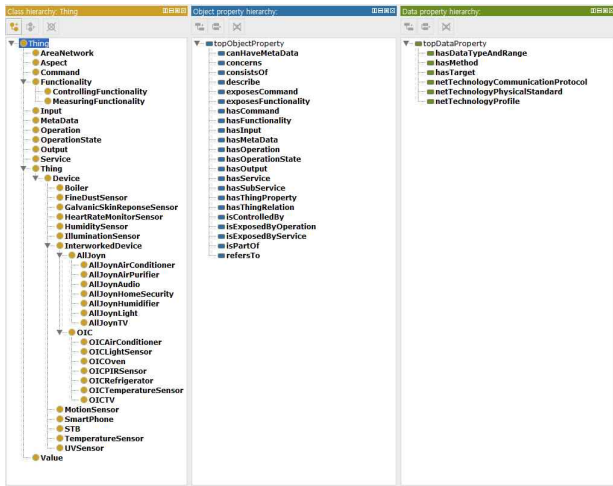


Fig. 4. Implementation of virtual objects (VO) with protégé

그림 4에서, 왼쪽부터 클래스, 오브젝트 프로퍼티, 데이터 프로퍼티를 나타내며, 다양한 IoT 디바이스를 식별하기 위하여 약 50여개의 센싱 디바이스와 40여개의 액추에이터를 표현할 수 있게 개발 되었다. 또한 센싱 디바이스의 경우, 수집되는 데이터의 단위를 표현할 수 있도록 개발 되었으며, 센싱 디바이스 및 액추에이터 디바이스 모두 기능에 대한 분류, 기능을 실행하기 위한 파라미터 등을 표현할 수 있도록 개발되었다.

### 2. CVO Implementation

CVO를 개발하기 위하여 VO 개발과 동일하게 프로티지를 통하여 개발하였다. 그림 5는 프로티지를 통한 CVO의 구현 결과를 보인 것이다.

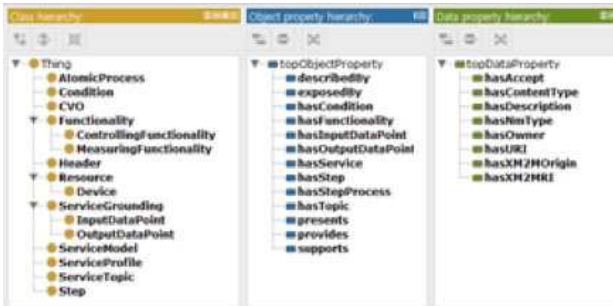


Fig. 5. Implementation of composite virtual objects (CVOs) with protégé

그림 5에서, 왼쪽부터 클래스, 오브젝트 프로퍼티, 데이터 프

로퍼티를 나타내며, VO의 리소스 (즉, Device 클래스, Functionality 클래스)를 재활용하여 복합 서비스 구성 시 해당 리소스를 참조할 수 있도록 개발 되었다. 또한, 실제 서비스 구동 시 개별 VO에 접근할 수 있는 리소스의 URI, 파라미터, 프로토콜 등을 표현할 수 있도록 개발 되었다.

### 3. Web-based Composition Tool

본 절에서는 개발자가 동적 서비스 구성 방법을 통하여 개발하기 위한 웹 기반 편집 툴에 대하여 기술한다. 웹 기반 편집 툴은 개발자가 쉽고 편리하게 이용할 수 있도록 그래픽 유저 인터페이스 (GUI, Graphic User Interface)를 지원한다. 또한 서비스 구성 시 VO를 활용하여 각 단계별 소단위 서비스 (Atomic Service)를 구성할 수 있으며, 최종적으로는 소단위 서비스를 기반으로 복합 서비스를 구성할 수 있게 한다.

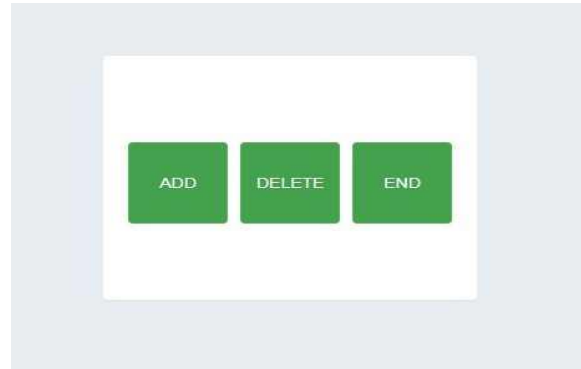


Fig. 6. A snapshot of start menu

이를 위하여, 개발자는 그림 6과 같이 서비스 구성 시 ADD (추가), DELETE (삭제), END (종료) 중 하나를 선택하여 서비스 구성을 시작한다. 여기에서 ADD는 서비스 단계를 추가하는 것으로 개별 VO를 기반으로 각 단계의 서비스를 정의 할 수 있다. DELETE는 정의된 서비스 단계를 삭제하는 기능을 가지며, END는 서비스 구성을 종료할 때 사용한다. 개발자가 개발을 완료한 후 END 버튼을 선택하게 되면, 구성된 서비스는 XML 형태로 IoT 플랫폼에 전달되며, 이를 기반으로 IoT 플랫폼에서는 CVO 템플릿을 통하여 신규 CVO를 생성하게 된다. 동시에, 해당 CVO를 기반으로 IoT 플랫폼은 물리 공간에 있는 IoT 디바이스를 제어한다.

서비스 생성을 위하여 단계 추가 (ADD)를 선택한 다음 개발자는 구성하고자 하는 서비스 단계에서 필요한 기능을 그림 7과 같이 선택할 수 있다. 이때, 추가되는 기능은 그림 7과 같이 일곱가지 기능 (Measuring, Measuring Condition, Time Condition, Situation Condition, Emotion Condition, Controlling, Controlling Condition)을 지원한다.

Measuring, Controlling의 경우, IoT 디바이스에서 지원하는 조회 기능과 제어기능을 나타내며, 나머지 다섯가지의 기능은 사전 조건 또는 사후 조건을 위하여 사용한다.

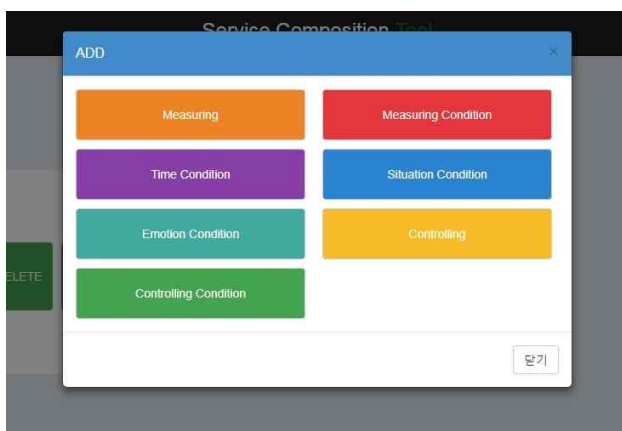


Fig. 7. a snapshot of selecting function

단계별 제공 기능 선택 후 개발자는 해당 기능을 포함하는 IoT 디바이스의 목록을 자동으로 확인할 수 있다. 이러한 이유는 IoT 디바이스의 프로파일 정보를 기반으로 만들어진 VO에서 기능의 의미를 추출하고, 해당 기능을 포함하는 디바이스를 최종적으로 선택할 수 있기 때문이다. 그림 8은 웹기반 편집 틀을 통하여 구성된 예시를 보인 것이다. 그림 8에서 보인 것과 같이 개발자는 단계별 서비스를 구성한다. 예를 들면, STEP1에서는 특정 기기의 조회 기능을 이용하여, 기기의 상태를 조회하는 서비스를 구성 (대상 기기의 상태 감지)하고, 조회된 값을 이용하여 STEP2에서 사전 조건 값 (예: 대상 기기의 상태 값)이 변하면 STEP3으로 이동을 정의할 수 있다. 마지막으로 STEP2에서 정의한 사전조건이 YES일 때 수행할 내용을 다음 STEP3에서 정의하여, 하나의 복합 서비스를 구성할 수 있다.

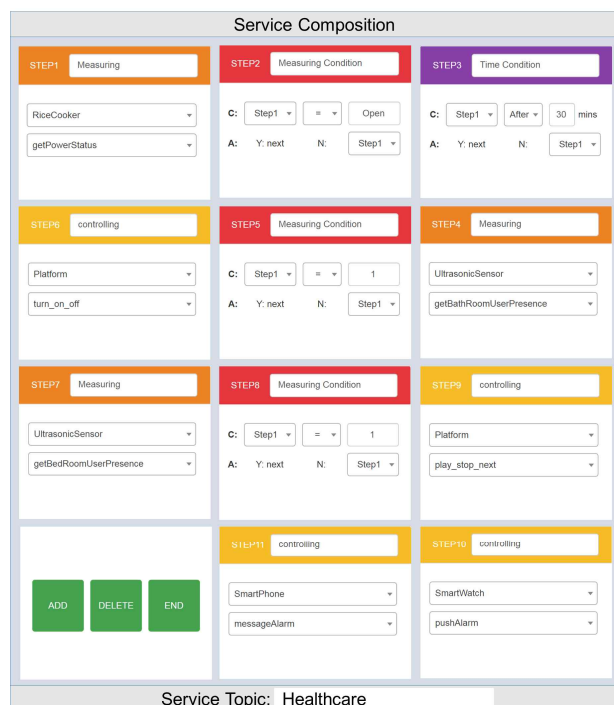


Fig. 8. An example of service composition

## V. Conclusions and Future Works

본 논문에서는 이중 IoT 시스템들을 이용하여 동적 서비스를 구성하기 위한 방법을 제시하고, 이를 기반으로 웹 기반 편집틀을 개발하여 실제 시스템에 적용 시 효율성을 검증하였다. 제안한 동적 서비스 구성 방법은 IoT 플랫폼에 등록된 IoT 디바이스 리소스를 시맨틱 기반의 가상 웹 오브젝트 (VO: virtual object)로 생성하고, 최종적으로는 개발자가 서비스를 구성할 때 웹 오브젝트 접근을 통하여 쉽고 편리하게 IoT 서비스를 구성할 수 있는 환경을 제공하였다. 또한, IoT 서비스에서 주요 사용되는 기능들을 사전에 정의하고, 해당 기능에 적합한 IoT 기기를 시맨틱 기술을 이용하여 자동 검색함으로써, 서비스 구성 시 효율성과 편리성을 증대시켰다. 마지막으로, 제안한 방법으로 구성된 서비스는 복합 가상 오브젝트 (CVO: composite virtual object)로 정의되어, 기존 서비스를 편집하거나 또는 재 활용할 수 있는 환경을 제공하였다. 본 논문에서 제안한 동적 서비스 구성 방법은 특정 산업 영역으로 활용이 제한되지 않으며, 이중산업 분야 간에 시스템을 동적으로 구성하여 연결하는 융합 서비스 개발에 활용성이 매우 높을 것으로 예상할 수 있다. 향후 본 논문에서 제안한 서비스 컴포지션 관련 기술들의 시맨틱 온톨로지와 IoT 플랫폼 관련 표준화 활동을 통해 기술의 활용성과 시스템 확장성 증대할 필요가 있다.

## REFERENCES

- [1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, Vol. 54, No. 15, pp. 2787-2805, 2010.
- [2] M. Ryu, J. Kim, and J. Yun. "Integrated semantics service platform for the Internet of Things: A case study of a smart office," *Sensors* Vol. 15, No. 1, pp. 2137-2160, 2015.
- [3] M. Ryu, J. Yun, T. Miao, I.-Y. Ahn, S.-C. Choi, and J. Kim, "Design and Implementation of a Connected Farm for Smart Farming System," in *Proceedings of the IEEE Sensors*, pp. 1724-1728, November 1-4 2015.
- [4] D. Bandyopadhyay and J. Sen, "Internet of things: Applications and challenges in technology and standardization," *Wireless Personal Communications*, Vol. 58, No. 1, pp. 49-69, 2011.
- [5] R. V. Kranenburg, E. Anzelmo, A. Bassi, D. Caprio, S. Dodson, and M. Ratto, "The internet of things," In *Proceedings of the 1st Berlin Symposium on Internet and Society*, Germany, pp. 25-27, October 2011.
- [6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural

- elements, and future directions,” *Future Generation Computer System*, Vol. 29, No. 7, pp. 1645–1660, 2013.
- [7] V. Gazis, K. Sasloglou, N. Frangiadakis, P. Kikiras, A. Merentitis, K. Mathioudakis, and G. Mazarakis, “Architectural blueprints of a unified sensing platform for the Internet of Things,” In *Proceedings of the 22 International Conference on Computer Communications and Networks (ICCCN 13)*, pp. 1–5, 2013.
- [8] R. Perrey and M. Lycett, “Service-oriented architecture,” in *Proceedings of Applications and the Internet Workshops*, pp 116–119, 2003.
- [9] J. Brønsted, K. M. Hansen, and M. Ingstrup, “Service Composition Issues in Pervasive Computing,” *IEEE Pervasive Computing*, Vol. 9, No. 1, pp. 62–70, January–March 2010.
- [10] S. Kalasapur, M. Kumar, and B. A. Shirazi, “Dynamic Service Composition in Pervasive Computing,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, No. 7, pp. 907–918, July 2007.
- [11] M. Kumar, B. A. Shirazi, S. K. Das, M. Singhal, B. Sung, and D. Levine, “Pervasive Information Communities Organization (PICO): A Middleware Framework for Pervasive Computing,” *IEEE Pervasive Computing*, Vol. 2, No. 3, pp. 72–79, 2003.
- [12] Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo, S. Bourne, and X. Xu, “Web services composition: A decade’s overview,” *Information Sciences*, Vol. 280, pp. 218–238, October 2014.
- [13] D. Guinard, V. Trifa, and E. Wilde. “A resource oriented architecture for the web of things.” *Internet of Things (IOT)*, pp. 1–8, 2010.
- [14] oneM2M: <http://www.onem2m.org/>.
- [15] oneM2M, TS-0012, “oneM2M Base Ontology,” 2016.
- [16] M3-lite ontology: <http://lov.okfn.org/dataset/lov/vocabs/m3lite>.
- [17] J. Kim, S.-C. Choi, I.-Y. Ahn, N.-M. Sung, and J. Yun, “From WSN towards WoT: Open API Scheme Based on oneM2M Platforms,” *Sensors*, Vol. 16, No. 10, pp. 1645, October 2016.
- [18] protege: <http://protege.stanford.edu/>.

## Acknowledgement

Thanks to Nak-Myoung Sung and Il-Yeup Ahn from Korea Electronics Technology Institute (KETI) for assistance with oneM2M standard and platform analysis, and Sheik Mohammad Mostakim Fattah from University of Sydney for assistance with the service composition tool development.

## Authors



Minwoo Ryu received the B.S. degree in Internet Information Processing from Yeosu Institute of Technology in 2007, and his M.S. and Ph.D. in Computer Science from Kwangwoon University, Korea, in 2009 and 2012, respectively. From 2011 to 2016, he worked as a senior researcher at Korea Electronics Technology Institute (KETI). He is now a research scientist at KT R&D Center. He is interested in IoT, ontology, loV and cognitive computing.



Jaeseok Yun received his B.S. degree in Electronics Engineering from Chonnam National University in 1997. He also earned his M.S. and Ph.D. degrees in Mechatronics from Gwangju Institute of Science and Technology (GIST) in 1999 and 2006, respectively. He is now an assistant professor with the Department of Internet of Things at Soonchunhyang University. Prior to his current position, he worked as a senior researcher with IoT Platform Research Center at Korea Electronics Technology Institute (KETI) from 2009 to 2016. He also worked as a postdoctoral research scientist with the Ubiquitous Computing Research Group in the School of Interactive Computing at Georgia Institute of Technology, GA, USA from 2006 to 2009. His research interests include ubiquitous computing, Internet of Things, machine learning, and intelligent systems.