

마이크로 그리드용 공통플랫폼 서비스 개발 도구

이현희¹ · 유대상^{2*}

Service Development Tool for Micro Grid Common Platform

Hyun-Hee Lee¹ · Dae-Sang You^{2*}

¹Elcomtech Co., Ltd., #262, Singalma-ro, Seo-gu, Daejeon 302-828, Korea

^{2*}Elcomtech Co., Ltd., #262, Singalma-ro, Seo-gu, Daejeon 302-828, Korea

요 약

마이크로 그리드는 기존의 대규모 전력망이 가진 여러 문제점을 해결할 수 있는 새로운 전력망 개념으로 세계적으로 국가차원에서 연구 및 개발 중이다. 이러한 배경에서 마이크로 그리드 내 장치 및 시스템 간 상호운영기술 및 신규시장 Enabler 확보를 위해 마이크로 그리드용 공통 플랫폼이 개발 중으로 이는 Third-Party 사업자에게 개방형 API 기능을 통해 전력정보를 제공하여 비즈니스 모델을 지원하도록 한다. 마이크로 그리드용 공용 플랫폼에 접속하여 전력 어플리케이션을 개발하고자하는 사업자는 서비스 개발 및 배포를 지원하는 통합 개발 환경이 필요하다. 이에 본 논문은 마이크로그리드용 공통플랫폼의 어플리케이션 개발 및 배포를 위한 PaaS(Platform as a Service) 형태의 서비스 개발도구를 제안한다.

ABSTRACT

Micro Grid is a new concept of electrical grid to solve several problems of Mega-Grid and it is under research and development at the national level around the world. With this background, Micro Grid Common Platform has been developed for interoperability technology of devices-systems in Micro Grid and securing enabler for new market. For supporting business models, Micro Grid Common Platform provides electric power information to Third-Party licensee via Open API function. Licensee, who want to develop electric power application by connecting to Micro Grid Common Platform, needs IDE(Integrated Development Environment) for service development and distribution. The paper proposes a service development tool for developing applications of Micro Grid Common Platform as a PaaS (Platform as a Service).

키워드 : 마이크로 그리드, 클라우드서비스, 플랫폼

Key word : Micro Grid, Cloud Service, PaaS

Received 28 April 2015, Revised 14 May 2015, Accepted 26 May 2015

* Corresponding Author Dae-Sang You(E-mail:dseyou@elcomtech.co.kr, Tel:+82-42-824-0120)
Elcomtech Co., Ltd., #262, Singalma-ro, Seo-gu, Daejeon 302-828, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2015.19.6.1455>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

1.1. 연구 배경 및 목적

마이크로그리드란, 소규모 독립형 전력망 내에서 전력정보를 공유하여 에너지 효율을 최적화하는 지능형 전력망 기술로서 일정지역 내에서, 풍력태양광 발전 등 신재생에너지원과 에너지 저장장치(ESS, Energy Storage System) 등을 에너지 관리시스템(EMS, Energy Management System)으로 제어하여 외부의 전력망에 연결하거나 독립적으로 운전할 수 있는 소규모 전력망이다. 마이크로 그리드는 인구가 밀집한 도심지역에서 발전자원과 함께 수요자원 안정성과 효율성을 극대화할 수 있는 기술이다.

에너지 분야 컨설팅 기업인 Navigant Research의 2013년 보고서에 따르면 현재 세계적으로 계획 또는 개발·운영 중인 마이크로그리드 관련 프로젝트는 405개에 달하며 시장 규모는 2013년 기준으로 연간 83억 달러에서 2020년에는 약 200억 달러, 4,000MW로 연평균 30%씩 성장할 것으로 전망된다[1,3].

이러한 트렌드를 배경으로 마이크로그리드 내의 발전, 판매, 시장 등 다양한 영역의 시스템의 통합운영을 지원하고 장치 및 시스템 간 상호 운용성 제공 및 Third-party 사업자에게 개방형 API 기능을 통해 전력정보를 제공하여 비즈니스 모델을 지원하도록 하는 마이크로그리드용 공통플랫폼이 개발 중이다.

마이크로 그리드용 공통 플랫폼을 제공받는 사업자들은 플랫폼을 제공받아 이용할 때 사업자마다의 운영 환경이 다르므로 공통 플랫폼에서 제공하는 기본 서비스 외에 각 환경에 맞는 응용서비스를 개발하여 사용해야만 한다. 응용 서비스의 개발은 플랫폼 개발자 없이는 불가능한 작업으로 이를 위해 플랫폼 제공자가 사업자를 지원함에 있어 모든 사업자의 요구 사항을 충족시키기 어려울 뿐 아니라 적지 않은 비용이 발생 할 수 있다. 따라서 공통플랫폼을 제공할 때 사업자들이 서비스 개발자로 하여금 마이크로그리드용 공통플랫폼에 직접 접속하여 전력 어플리케이션을 개발하고 배포할 수 있도록 서비스 개발도구를 같이 제공할 필요가 있으며 이에 효과적으로 지원하기 위한 방법으로 어플리케이션 개발 자동화 틀을 PaaS (Platform as a Service) 형태의 클라우드 서비스로 운영 및 제공하면 비용절감과 편의성을 도모할 수 있다.

이러한 배경을 바탕으로 본 논문은, 마이크로그리드용 공통플랫폼의 어플리케이션 개발 및 배포를 위한 PaaS 형태의 서비스 개발도구를 제안한다.

1.2. 관련 기술 동향

마이크로그리드에 대한 실증적 연구는 EU, 일본, 미국 등 선진국을 중심으로 2000년대 초반부터 추진되어 왔으며 북미지역의 경우 200여개 이상의 프로젝트를 추진하고 있으며 캘리포니아 대정전 및 허리케인 ‘샌디’와 같은 자연재해 시 전력공급 신뢰도 확보 등을 위하여 실증사업을 추진하고 있다. 유럽 역시 마이크로그리드의 기술적인 요소에 대한 실증 및 상업적 요소 및 표준화 등을 위하여 실증사업을 추진 진행 중이며, 일본도 많은 섬들을 활용해 다양한 규모의 마이크로그리드 시스템을 실증하고 있다[3]. 우리나라의 경우 한전 전력연구원이 2010년 2월부터 3년 동안 200kW급 계통 연계형 마이크로그리드 실증 사이트를 구축하고, 요소기기 개발, 통합 기능시험, 운전성능 평가 등을 진행한 바 있다[2]. 클라우드 컴퓨팅 기술의 표준과 관련하여서 ITU-T(International Telecommunications Union Telecommunication)와 ISO/IEC JTC 1(Joint Technical Committee One)에서는 각각 SG13(Study Group 13)와 SC38(ISO 산하 정보통신분야 국제표준화 공동기술 위원회)에서 각자 고유의 클라우드 컴퓨팅 관련 표준을 제정하고 있고 또한 두 기구에서는 공동으로 CT(Collaborative Team)를 구성하여 클라우드 컴퓨팅에 관련된 표준을 작성 중이다. 국내에서는 TTA가 PG420, STC2/SPG21, PG414, PG421, PG424등을 통해서 클라우드 컴퓨팅의 표준화 작업을 진행하여 퍼스널 클라우드, 데스크톱 서비스, 모바일클라우드, 인프라서비스, 클라우드 기반 빅데이터 서비스, 스마트 사무환경, 미디어 서비스 등의 클라우드 기반 서비스들에 관련된 표준이 다수 제정되었다. 또한, 클라우드 컴퓨팅의 일반적인, 보안, 용어, 유즈케이스, SLA의 기반 표준들도 제정된 상태이다[4-8].

II. 개발 도구의 개요

2.1. 개발 도구의 목적 및 구성

본 개발도구는 마이크로그리드용 공통플랫폼에서

제공하는 OpenAPI를 이용하여 마이크로그리드 운영 및 관리 그리고 모니터링 서비스를 제공하는 웹기반 어플리케이션 개발환경과 실행환경을 제공한다. 이를 위해 본 개발도구는 크게 웹브라우저 상에서 구동되는 부분과 실행엔진에서 구동되는 부분으로 크게 나뉘어져 구성되어 있다. Open API를 등록하는 모듈 그리고 등록된 서비스를 콤포지션하는 부분이 있으며 이를 조합하여 매쉬업(Mash-up)하는 위젯 및 어플리케이션 빌더가 있다. 저장된 어플리케이션을 수행하는 브라우저와 실행엔진이 있으며 이에 사용되는 데이터베이스, Context Broker가 존재한다. 또한 백그라운드서비스를 제공하는 Task Manager가 제공되며 각종 스키마를 점검하는 점검기가 제공된다.

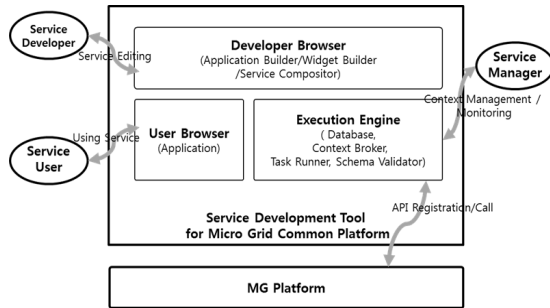


그림 1. 마이크로 그리드용 공통플랫폼 서비스 저작 도구
Fig. 1 Service Development Tool for Micro Grid Common Platform

- **Developer Browser** : 서비스 오케스트레이션 저작 도구, 위젯 생성기 및 어플리케이션 저작도구 서비스를 제공
- **User Browser**: 저장된 어플리케이션을 수행
- **실행엔진**: 저장된 서비스 및 위젯 어플리케이션을 수행하기 위한 저장소와 실행엔진 서비스를 제공

2.2. 개발 도구의 활용 시나리오

본 개발도구는 마이크로 그리드 공통 플랫폼에서 제공하는 서비스를 등록 및 관리하고 등록된 서비스를 매쉬업하여 또 다른 서비스를 제공한다. 서비스 형태는 크게 서비스를 단순 조합한 서비스 콤포지션과 서비스와 웹UI를 결합한 위젯 서비스 그리고 위젯서비스를 결합한 어플리케이션 형태를 동시에 제공한다. 이를 위해서 서비스 오케스트레이션 저작도구, 위젯 생성기 및

어플리케이션 저작도구 서비스를 제공하고 저작된 서비스 및 위젯 어플리케이션을 수행하기 위한 저장소와 실행엔진 서비스를 제공한다.

- **Open API 등록 관리** :관리자는 Context Broker에 접속하여 서비스 저작에 필요한 Open API 등록을 수행한다. Open API 등록을 요청받은 Context Broker는 MG 플랫폼(MG Platform)으로부터 Open API 버전 등의 정보를 얻어 와서 Context를 생성하여 저장한다.
- **서비스 저작** : 개발자가 서비스 저작 툴을 이용하여 서비스를 저작하고 저작한 서비스를 조회, 수정, 삭제한다. 개발자는 서비스 구현 및 관리를 위해 저작 툴을 제공받고 서비스 구현에 필요한 컨텍스트 정보를 조회하고 조합하여 서비스를 저작한다.
- **위젯 저작**: 개발자가 위젯 저작 툴을 이용하여 위젯을 구현하고 구현한 위젯을 조회, 수정, 삭제 및 테스트한다. 개발자는 위젯구현 및 관리를 위해 저작 툴을 제공받고 위젯구현에 필요한 서비스 정보를 조회하고 조합하여 위젯을 저작한다.
- **어플리케이션 저작**: 서비스 개발자가 어플리케이션 저작 툴을 이용하여 어플리케이션을 저작하고 저작한 어플리케이션을 조회, 수정, 삭제한다. 개발자는 어플리케이션 구현 및 관리를 위해 저작 툴을 제공받고 어플리케이션 구현에 필요한 위젯 정보를 조회하고 조합하여 어플리케이션을 저작한다.
- **서비스 실행**: 관리자는 Task Runner(Task Manager)가 백그라운드 서비스를 실행 시키도록 한다. 서비스가 실행되면 서비스를 구성하는 컨텍스트 호출을 Context Broker가 처리한다.
- **어플리케이션 실행**: 관리자는 서비스 실행엔진(Service Execution Engine)이 어플리케이션을 실행 시키도록 한다. 어플리케이션이 실행되면 어플리케이션을 구성하는 위젯, 서비스가 호출된다.
- **서비스/위젯/어플리케이션 저장소 관리**: 서비스 관리자는 저장소의 시스템 정보를 설정하고 서비스, 위젯, 어플리케이션 저장을 위한 환경 설정 정보를 설정한다.
- **서비스/어플리케이션 라이프사이클 관리**: 서비스 관리자는 백그라운드 서비스 및 어플리케이션을 실행 또는 종료 시킨다.
- **서비스/어플리케이션 모니터링**: 서비스 관리자는 실행 중인 백그라운드 서비스와 어플리케이션의 상태 및

Open API 호출 현황을 모니터링 한다.

III. 제안 도구 및 구조

3.1. 구성 요소별 기능모듈과 구조

본 논문에서 제안된 개발도구의 구성 요소 별 주요 기능모듈과 구조는 다음과 같다.

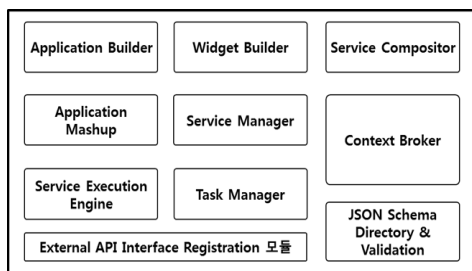


그림 2. 마이크로 그리드용 공통플랫폼 서비스 저작 도구 모듈 구성도

Fig. 2 A Structure of Service Development Tool for Micro Grid Common Platform

- External API Interface Registration 모듈 : 마이크로 그리드용 공통플랫폼에서 제공하는 Open API를 등록하는 모듈로서 외부에서 본 모듈의 인터페이스를 호출하여 Open API 접근정보를 기록한다. 기록된 정보는 서비스 콤포지터에서 호출하여 사용하며 이 때 필요한 정보(인터페이스 그룹, 인터페이스 명, 인터페이스 설명, 인터페이스 데이터 기술)을 기록하게 된다. 등록된 정보는 서비스 레포지토리에 등록된다.

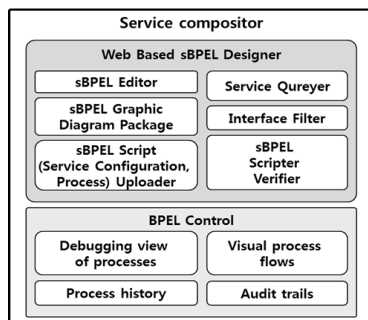


그림 3. Service Composer의 구조
Fig. 3 A Structure of Service Composer

- Service Composer: 서비스 조합을 위해서는 조합을 위한 웹기반 저작도구와 저장된 결과를 저장하는 모듈 서버 모듈 2개 레이어로 나누어지며 구성은 그림 3과 같다. 저작결과는 크게 서비스 기술서와 서비스 스펙으로 구성되며 서비스 스펙은 BPEL로 기술된다. (그림 3 참조)

- Widget Builder: Widget은 UI 및 서비스 통합한 S/W 컴포넌트이다. Widget을 저작하기 위해서는 Widget UI 컴포넌트를 관리하는 기능과 Widget UI와 서비스를 연동하는 저작도구 그리고 외부 연동 인터페이스 스펙을 저작 기록하는 세부 모듈로 나뉜다.(그림 4 참조)

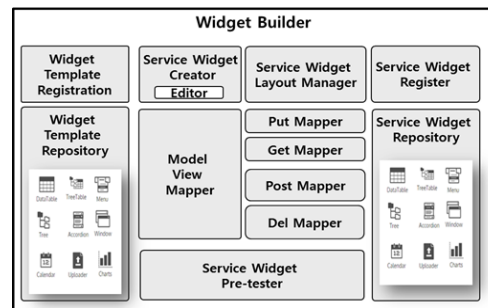


그림 4. Widget Builder의 구조
Fig. 4 A Structure of Widget Builder

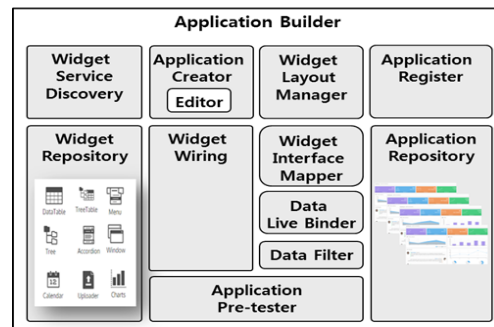


그림 5. Application Builder의 구조
Fig. 5 A Structure of Application Builder

- Application Builder: Widget을 조합하여 하나의 어플리케이션을 만드는 기능을 담당하는 모듈로 어플리케이션 조합을 하는 웹기반 저작 툴과 이를 저장 및 관리하는 서버 모듈 등으로 나누어진다.(그림 5 참조)
- Application Mashup 기능: Application Mashup은 웹 브라우저에서 수행된다. 크게 어플리케이션을 로드하

는 모듈과 어플리케이션에서 Widget간 연결된 인터페이스를 통해 Live Data Biding이 일어나는 부분으로 크게 구성으로 나누어진다. (그림 6참조)

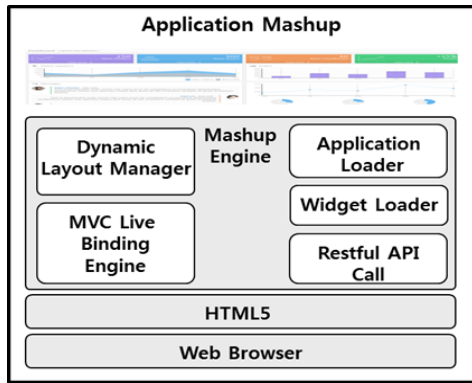


그림 6. Application Mash-up 구조
Fig. 6 A Structure of Application Mash-up

- Service Execution Engine: 위젯에 연결된 서비스를 실행하는 엔진으로서, 서비스를 로드하고 BPEL (Business Process Execution Language)을 파싱하여 수행하는 모듈로 구성된다. OpenAPI 호출과 Context Broker을 통한 동기 및 비동기 서비스를 제공한다. (그림 7참조)

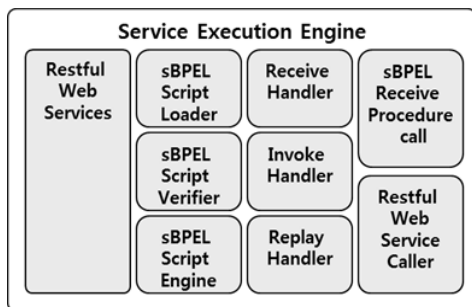


그림 7. Service Execution Engine 구조
Fig. 7 A Structure of Service Execution Engine

- Context Broker: Restful(REST:Representational State Transfer) 서비스를 통한 Context(데이터객체)를 생성 변경 파기를 하는 기능으로서, 서비스 및 어플리케이션 레포지토리 주소 관리 및 서비스 데이터를 관리한다. (그림 8참조)

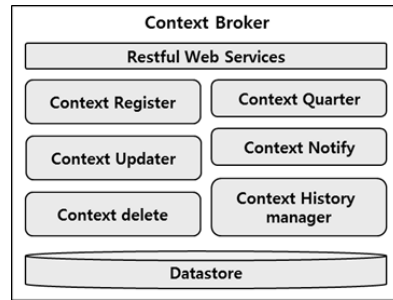


그림 8. Context Broker 구조
Fig. 8 A Structure of Context Broker

- Context Broker: Restful(REST:Representational State Transfer) 서비스를 통한 Context(데이터객체)를 생성 변경 파기를 하는 기능으로서, 서비스 및 어플리케이션 레포지토리 주소 관리 및 서비스 데이터를 관리한다. (그림 8참조)

- Task Manager: 서비스 중에 백그라운드로 표시된 서비스를 수행하는 모듈로 서비스를 fetch하는 모듈과 서비스를 수행하는 부분으로 나누어진다.
- Service Manager: MG서비스 개발도구 플랫폼에서 제공하는 서비스의 실행 상태를 관장하는 모듈로 크게 서비스 모니터 및 서비스 실행제어 기능으로 나누어진다.
- JSON Schema Directory and Validation: 스키마를 등록 관리하는 모듈로 스키마를 통한 JSON타입의 스키마를 점검하는 모듈로 구성된다.

3.2. 개발 도구 요소기술

본 개발 도구를 구현하기 위한 요소 기술의 설명은 다음과 같다.

- 서비스/위젯/어플리케이션 저작을 위한 그래픽 사용자 인터페이스 기술: Service Compositon은 그래픽 편집과 스크립트 편집 두 가지 방법을 이용하고 위젯과 어플리케이션 저작은 그래픽 편집 방법을 이용한다. 개발자가 서비스/위젯/어플리케이션의 그래픽 편집을 하려면 Datagramming Tool이 필요하고 이를 위해 그래픽 툴을 구현한다.
- 서비스/위젯/어플리케이션 저장: 서비스 스펙은 BPEL로 기술하며 그림 9,그림10와 같은 모델을 준수한다.

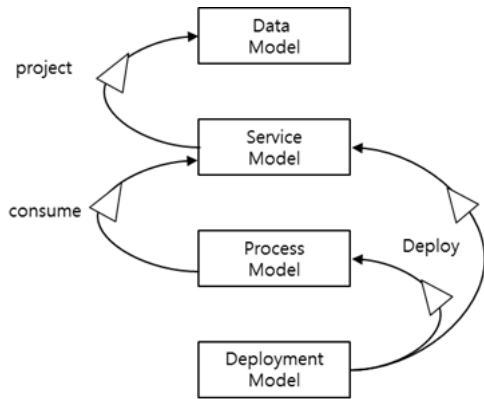


그림 9. 프로세스 정의 모델
Fig. 9 Process Definition Model

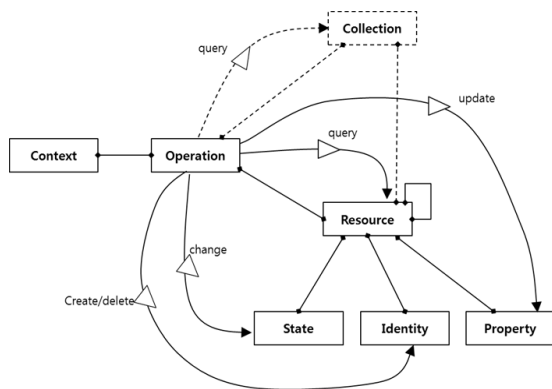


그림 10. 프로세스 동작 모델
Fig. 10 Process Operation Model

- 어플리케이션 실행: 본 개발 도구를 이용한 어플리케이션을 구현은 1. Open API 등록, 2. 컨텍스트 생성, 3. 컨텍스트를 조합하여 서비스 구현, 4. 서비스를 조합하여 위젯 구현, 5. 위젯을 조합하여 어플리케이션 구현하는 절차로 이루어진다. 구현된 어플리케이션은 서비스 관리자가 실행/종료에 대한 라이프사이클을 설정하고 서비스 실행엔진에서는 관리자가 설정한대로 어플리케이션을 실행/종료 시킨다. 어플리케이션이 실행되면 서비스 사용자가 어플리케이션에 접속하고 어플리케이션에 포함된 위젯, 서비스, 컨텍스트, Open API가 순차적으로 호출되어 MG Platform에서 제공하는 Open API 호출 결과가 어플리케이션에 반영되게 된다.(그림 11 참조)

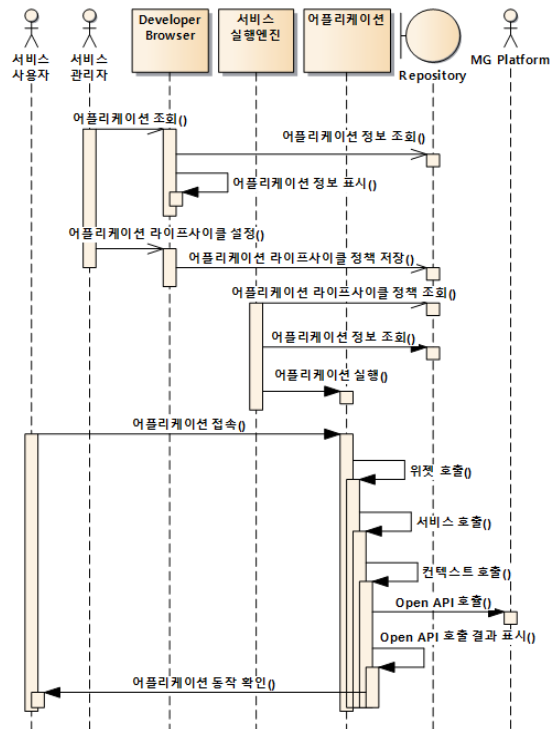


그림 11. 프로세스 동작 모델
Fig. 11 Process Operation Model

IV. 결론

본 논문은 마이크로그리드용 공통플랫폼의 어플리케이션 개발 및 배포를 위한 서비스 개발도구를 제안하였다. 제안된 개발 도구는 크게 Developer Browser, User Browser, 실행엔진 등으로 구성되며 Open API 등록 관리, 서비스/위젯/어플리케이션 저작, 서비스 실행, 어플리케이션 실행, 서비스/위젯/어플리케이션 저장소 관리, 서비스/어플리케이션 라이프사이클 관리, 서비스/어플리케이션 모니터링 기능을 수행한다. 이 개발도구를 통해 공통플랫폼을 제공받는 사업자는 보다 쉽게 필요한 응용서비스를 개발하고 활용할 수 있을 뿐 아니라 개발 도구가 클라우드 서비스 형태로 되어 시스템 운영 측면에서도 비용 절감과 편의성을 증대시킬 수 있다. 향후에는 본 논문에 기술한 개발도구를 구현하고 실증할 것이며 보안 기술을 접목하여 안정성을 더할 수 있도록 연구를 진행할 것이다.

본 개발도구는 현재 산업통상자원부 에너지기술개발사업의 일환인 “마이크로 그리드용 공통 플랫폼 기술 개발”로 개발 중에 있으며 활용 시 마이크로 그리드 공통 플랫폼의 서비스/애플리케이션 개발 증대 및 배포 자동화가 기대되며 이는 마이크로 그리드용 공통 플랫폼에 접속하는 사업자들에게 편리한 통합개발환경을 제공함으로써 보다 효율적인 지원이 가능할 것이다.

ACKNOWLEDGMENTS

This work was supported by Energy Technology Program of the Ministry of Trade, Industry & Energy.

REFERENCES

[1] “Microgrids”, Navigant Research, 2013.

[2] “Microgrid”, *Journal Of Electrical World/Korea Electric Association*, Vol 448,66-67,2014.04

[3] Sung-Man Kim, “Application and Case Study of Microgrid Technology”, *Journal Of Electrical World/Korea Electric Association*,Vol 455, 37-44,2014.11

[4] M.-H. Oh, K.C. Lee, S.W. Kim, H.-Y. Kim,W. Choi, “Trends in Domestic and International Standardization of Cloud Computing”, *ETRI/Electrics and Telecommunications Trends*, Vol 148, 59-71, 2014

[5] “Guideline for Resource Provision and Status Management in Cloud Infrastructure Service,” TTAK.KO-10.0465, Telecommunications Technology Association (TTA), 2010.

[6] “Framework of Cloud Desktop as a Service,” TTAK.KO-10.0468, TTelecommunications Technology Association (TTA), 2010.

[7] ISO/IEC JTC1/SC39 Substantiality for and by Information Technology. Available: http://www.iso.org/iso/home/standards_development/list_of_iso_technical_committees/iso_technical_committee.htm?commid=654019.

[8] ICWG/2302 WG - Intercloud WG (ICWG) Working Group. Available: http://standards.ieee.org/develop/wg/ICWG-2302_WG.html



이현희(Hyun-Hee Lee)

2009년 ~ 현재 ㈜엘컴텍 과장
 2012년 충남대학교 전자전파정보통신공학과 공학석사
 2005년 평생교육진흥원 컴퓨터공학과 공학사
 ※관심분야: 디지털 방송 · 통신 시스템, 가상화, 3D 그래픽스



유대상(Dae-Sang You)

2009년 ~ 현재 공주대학교 대학원 컴퓨터공학과 박사과정
 2009년 ~ 현재 ㈜엘컴텍 기술이사
 2004년 충남대학교 정보통신공학과 공학석사
 1998년 충남대학교 수학과 이학사
 ※관심분야: 디지털 방송 · 통신 시스템, 가상화, 3D 그래픽스, 영상처리