

## 로봇 소프트웨어 개발을 위한 클라우드 기반 통합 개발 환경

# A Cloud-based Integrated Development Environment for Robot Software Development

윤재훈, 박홍성\*

(Jae Hoon Yoon<sup>1</sup> and Hong Seong Park<sup>1,\*</sup>)

<sup>1</sup>Dept of Electronic and Electrical Engr. IT College, Kangwon National University

**Abstract:** Cloud systems are efficient models that can utilize various infrastructures, platforms, and applications regardless of the type of clients. This paper proposes a cloud-based integrated development environment (IDE) for robot software development which would make software development easier. The proposed system provides robot simulation to test the robot HW modules or robot systems for development and testing of software operating in a robot system with two or more different operating systems (OS) such as Windows, Linux, and real-time OS. This paper implements and evaluates the proposed system using OPRoS [33].

**Keywords:** development environment, simulation environment, robot cloud

### I. 서론

로봇, PC와 패드(PAD) 등과 같은 클라이언트의 자원에 대한 제약 없이 SW를 개발하고 서비스를 활용하도록 지원하는 클라우드[1]는 미래 컴퓨팅 산업의 주요 기술로 평가되면서 현재 많은 연구가 이루어지고 있다[2-9]. 사용자는 로봇, 데스크톱, 노트북, 휴대용 기기 등 다양한 클라이언트를 이용해 작게는 데이터의 저장에서 크게는 응용 프로그램의 구동에 의한 서비스까지 서버를 통해 처리하는 클라우드 서비스를 제공받는다.

클라우드 서비스는 크게 메모리, 서버 등의 자원을 제공하여 주는 IaaS (Infrastructure as a Service), 응용 SW를 제공하여 주는 SaaS (Software as a Service), 개발 플랫폼을 구축할 필요 없이 개발 시 필요한 것들을 웹에서 쉽게 활용할 수 있게 하는 PaaS (Platform as a Service)로 크게 나눌 수 있다[1]. 특히 PaaS는 SaaS의 개념을 개발 플랫폼에 확장한 방식으로 생각할 수 있다.

현재 IT 분야에서는 IaaS, SaaS, PaaS 분야에서 다양한 응용을 제공하고 있지만, 로봇분야에서는 이러한 연구가 적고 실제로 대부분의 로봇분야의 클라우드 서비스 연구는 SaaS 분야로 제한되고 있다[3-6]. 로봇은 입력된 환경 데이터를 클라우드로 보내 알맞게 처리한 후 처리 결과를 클라우드로부터 받아 실행하는 역할을 한다. 이러한 서비스에는 복잡한 모션 생성, 환경 인지, 네비게이션 등의 서비스가 있을 수 있다. 그러나 이러한 서비스 로봇의 개발환경을 위한 PaaS 관련 연구는 존재하지 않는다.

IT 분야의 PaaS와 관련된 예들로는 다음과 같은 것들이 있다: 코딩, 디버그, 컴파일 및 배포 등의 프로그램 개발에 관련된 작업들을 하나의 SW 안에서 처리하는 통합 개발 환경(IDE: Integrated Development Environment) 서비스[7-9], 클라우드 서비스를 활용하는 API를 제공하고 이 API들을 사용하여 개발을 지원하는 서비스이다. 본 논문의 연구 분야로 PaaS 중 통합개발환경 서비스 분야로 제한한다.

이러한 IT 분야의 클라우드 기반 통합 개발 환경 서비스들을 로봇 분야에 적용하는 것은 힘들다. 이는 로봇 분야에서 클라우드 기반 통합개발환경 서비스는 다음과 같은 중요한 요구사항들을 만족하여야하기 때문이다.

- 로봇 SW 테스트 시 HW 부분과 연동되는 방법 제공
- 한 로봇에는 다중 보드 및/혹은 두 종류 이상의 운영체제를 사용 가능

첫 번째 요구사항은 로봇 소프트웨어는 하드웨어와 밀접하게 관련되어 있다는 사실로부터 나온 것이며, 개발된 소프트웨어의 동작을 테스트하기 위해서는 시뮬레이터 또는 해당 소프트웨어에서 필요로 하는 하드웨어가 필요하기 때문이다. 즉, 개발된 로봇 소프트웨어의 동작을 테스트하기 위한 클라우드 환경에서 실제 HW 연동 서비스 혹은 HW 시뮬레이션 서비스를 제공하여야 한다. 또한 두 번째 요구사항은 로봇은 대부분이 하나의 운영체제를 가지고 동작하지만 2개 이상의 이중 운영체제상에서 동작하는 경우도 존재하기 때문에 필요하다.

특히 클라우드 기반 통합 개발 환경 서비스는 로봇 소프트웨어 개발을 위해 네트워크 접속장치 기능을 가진 단말기(예: 로봇 혹은 데스크톱)를 클라우드 서버에 연결시켜 사용하게 한다. 이 때 단말기는 개발과 관련 모든 서비스는 클라우드에서 제공되기 때문에 많은 리소스를 가질 필요가 없는 신(thin) 클라이언트라 한다. 통합 개발 환경의 컴파일, 디버그, 그리고 개발 산출물의 저장과 함께 GUI를 모두 클

\* Corresponding Author

Manuscript received May 26, 2014 / revised November 20, 2014 / accepted December 11, 2014

윤재훈, 박홍성: 강원대학교 전자통신공학과

(iicjaehoon@kangwon.ac.kr/hspark@kangwon.ac.kr)

※ 본 논문은 산업통상자원부에서 지원하여 연구하였음.

라우드 서버에서 실행하며 실행되는 통합 개발 환경의 화면을 클라이언트에 제공한다. 로봇 시뮬레이터 또한 같은 방식으로 구동하고 사용자는 클라우드 내 여러 플랫폼에 접근할 수 있으며 다양한 플랫폼을 기반으로 개발된 로봇 소프트웨어를 연동하여 동작시키고 시뮬레이션 할 수 있다.

제안하는 로봇용 클라우드 기반 통합개발환경 서비스는 이러한 2가지 요구사항들을 모두 만족하면서 로봇 소프트웨어의 개발을 쉽게 하도록 도와준다. 특히 첫 번째 요구사항을 만족하기 위해 HW 동작을 위하여 이에 대한 로봇 시뮬레이션을 클라우드에서 제공하여 개발된 소프트웨어의 동작 테스트를 클라우드 내에서 지원하도록 클라우드 시스템을 구성한다. 또한 여러 운영체제를 가진 로봇 시스템인 경우 시뮬레이션을 위해서는 여러 운영체제를 포함하는 인프라스트럭처와 복잡한 환경설정이 필요한데 제안하는 시스템은 윈도우즈와 리눅스를 가상기계(virtual machine) 상에 구현하고 자동화된 환경 설정을 통해 이를 쉽게 활용할 수 있도록 지원한다.

본 논문에서는 클라우드에서 로봇 소프트웨어 개발을 쉽게 할 수 있도록 로봇용 클라우드 기반 통합개발환경 서비스를 제공하는 시스템을 제안한다. 제안하는 시스템은 로봇 소프트웨어를 클라우드 환경에서 개발할 수 있도록 통합개발환경과 로봇 시뮬레이터 서비스를 제공하고 다수의 이종/동종 운영체제가 동작하는 멀티 플랫폼을 기반으로 로봇 SW를 개발 할 수 있다. 또한 OPRoS [10,12-14]를 활용하여 제안된 시스템을 검증하였다.

II 장에서는 제안하는 시스템의 구성을 설명하고, III 장에서 제안된 시스템에 대한 구현 및 평가하며, IV 장에서 결론을 맺도록 한다.

**II. 클라우드 기반 로봇 개발 환경**

본 논문에서 제안하는 로봇 소프트웨어 개발을 위한 클라우드 환경의 전체 구성은 그림 1과 같다. 클라이언트, 인증 및 관리 시스템, 저장소, 그리고 클라우드 환경을 구성하기 위한 인프라스트럭처와 플랫폼을 포함하는 클라우드 플랫폼으로 구성되었다. 클라우드 플랫폼 내에 사용자에게 제공하기 위한 통합 개발 환경 및 시뮬레이터가 포함되며 이 플랫폼과 클라이언트가 연결되어 사용자에게 서비스를 제공한다.

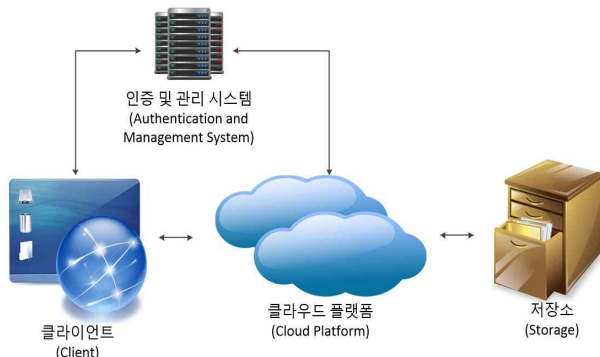


그림 1. 로봇 소프트웨어 개발을 위한 클라우드 아키텍처.  
Fig. 1. Cloud architecture for robot software development.

클라이언트의 리소스를 최소화한 신 클라이언트와 통합 개발 환경, 그리고 로봇 시뮬레이터를 연결하기 위해 원격 데스크톱 공유 시스템(remote desktop sharing)을 이용한다. 원격 데스크톱 공유 시스템은 VNC, Citrix의 Metaframe, 그리고 마이크로소프트의 RDP (Remote Desktop Protocol) 등이 존재하며 클라우드 환경 내에서 동작하는 다양한 응용에 대한 디스플레이를 클라이언트로 전송 할 수 있다.

인증 및 관리 시스템은 사용자 정보를 관리하고 사용자와 통합 개발 환경 및 로봇 시뮬레이터가 구동되는 서버와의 연결을 관리하는 시스템이다. 클라이언트로부터 연결 요청과 함께 데이터를 전송 받을 수 있어야 하며 클라우드 플랫폼 내에 존재하는 서버들을 모니터링 하여 클라이언트의 연결 요청에 대한 응답으로 정확한 서버 정보를 전송해야 한다. 클라이언트의 연결 요청 시 인증 및 관리 시스템에게 사용자 정보와 함께 원하는 플랫폼 정보를 입력받고 인증 및 관리 시스템은 그에 해당하는 서버를 클라우드 플랫폼 내에서 찾아 클라이언트에게 전송한다. 클라이언트는 전송받은 서버 정보를 이용해 해당 서버와 연결한다.

저장소는 사용자의 개발을 돕기 위한 다양한 자원을 저장하고 사용자의 개발 산출물을 저장한다. 사용자의 개발을 돕기 위한 자원은 로봇 소프트웨어를 개발하기 위한 소프트웨어 컴포넌트, 완성된 로봇 응용과 로봇 시뮬레이터에서 활용하기 위한 로봇모델, 환경모델 등을 포함한다. 이러한 자원은 서비스 제공과 함께 미리 준비되어 사용자들에게 제공하여 하며 사용자로부터 개발된 자원들도 업로드 할 수 있어야 한다. 사용자 개인 저장소에는 비정상적인 연결 종료료를 포함한 모든 연결 해제 시점까지의 개발 산출물이 저장되어야 한다. 또한 사용자 개인 저장소는 다른 사용자의 접근이 제한되어야 한다. 저장소는 클라우드 플랫폼 내에 사용자가 접속한 서버와 연결되고 사용자의 서버 연결 종료 시 해당 서버와의 연결이 종료된다.

제안하는 클라우드 플랫폼은 그림 2와 같이 구성된다. 하드웨어를 가상화하기 위한 하이퍼바이저(Hypervisor) 위에 여러 가상 머신이 존재하며 그 가상 머신 위에 다양한 운영체제가 존재한다. 하이퍼바이저는 대표적으로 VMware의 ESX Server, Citrix의 XenServer, 그리고 마이크로소프트의 하이퍼-V 등이 존재한다. 하이퍼바이저를 통해 가상화된 서버들이 사용자와 연결되는 서버들이며 다양한 개발환경 및 애플리케이션 구동 환경을 제공하기 위해 여러 OS를 포함

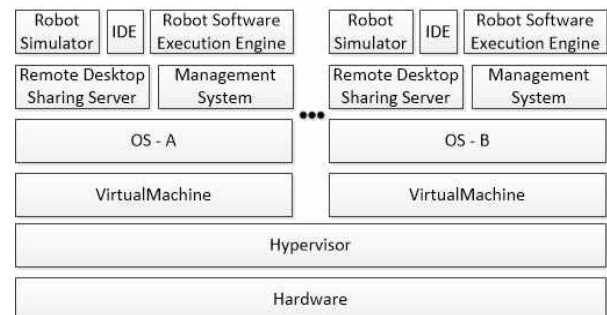


그림 2. 제안하는 클라우드 기반 통합개발환경.  
Fig. 2. The proposed cloud-based IDE.

한다. 이 OS들에 클라이언트와 클라우드 환경 내 서버의 디스플레이 연결을 위한 원격 데스크톱 공유 서버가 존재하고 인증 및 관리 시스템과 각 서버의 연결 및 해당 서버를 관리하기 위한 Management System이 존재한다. 이렇게 구축된 환경에 로봇 시뮬레이터와 로봇 개발을 위한 통합 개발 환경이 구성되며 사용자의 선택에 따라 다양한 종류의 플랫폼을 기반으로 로봇 소프트웨어를 개발 할 수 있다. 개발된 로봇 소프트웨어는 로봇 소프트웨어 실행 엔진을 통해 구동되고 하나의 클라이언트와 서로 다른 플랫폼의 여러 서버가 연결 될 수 있다. 또한 각 서버의 실행엔진 간의 통신을 통해 다중 보드 혹은 한 종류 이상의 운영체제를 사용하는 로봇 소프트웨어의 구동을 지원한다.

### III. 클라우드 기반 통합개발환경의 구현

제안하는 시스템은 그림 3과 같이 구현되었다. Intel Core 2 Duo 프로세서와 SATA 하드디스크를 이용해 스토리지와 물리 서버를 구축하고 로컬 네트워크를 통해 각 서버를 연결 하였다. 하이퍼바이저는 Citrix XenServer를 사용하였으며 물리서버를 가상화하여 클라우드 플랫폼을 위한 인프라스트럭처를 구축하였다. XenServer가 설치된 물리 서버와 XenServer를 통해 가상화된 가상 서버는 Citrix에서 제공하는 XenCenter을 이용해 클라우드 인프라스트럭처 내에 존재하는 물리 서버 및 가상 서버를 관리 할 수 있다.

구축된 클라우드 인프라에 OPRoS 로봇 플랫폼을 적용하

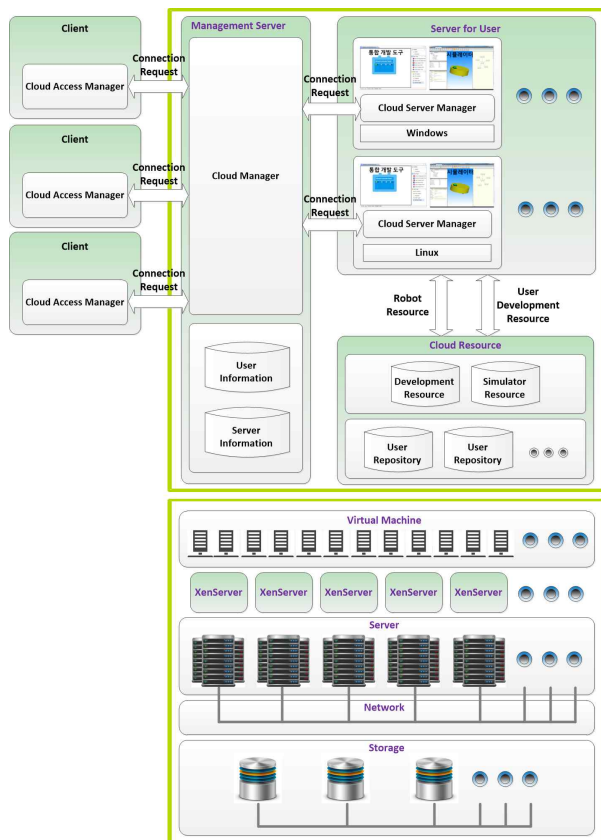


그림 3. 클라우드 기반 통합개발환경의 시스템 구성.  
Fig. 3. System configuration for cloud-based IDE.

여 그림 4와 같이 클라우드 플랫폼을 구축하였다. 다수의 가상서버에 일반 windows, linux 및 RTX와 Ubuntu RealTime이 패치된 OS의 이미지를 바탕으로 다양한 개발 환경을 구축하였다.

원격 데스크톱 공유 시스템은 Windows 서버의 경우 RDP와 VNC를 모두 지원하고 Linux는 VNC만 지원한다. 그림 4의 Management System의 역할을 CSM (Cloud Server Manager)가 수행하고 해당 서버의 관리 및 사용자와 해당 서버의 연결을 돕는다. 로봇 시뮬레이터는 OPRoS 로봇 시뮬레이터를 사용하고 개발도구는 OPRoS IDE를 사용하며 로봇 소프트웨어 실행 엔진은 OPRoS 컴포넌트 엔진을 사용한다. OPRoS 컴포넌트를 기반으로 구현된 로봇 소프트웨어들은 모두 OPRoS 컴포넌트 엔진을 통해 구동된다.

OPRoS 컴포넌트를 기반으로 구현된 로봇 소프트웨어와 OPRoS 로봇 시뮬레이터의 연동은 그림 5에 나타나 있다. 로봇 소프트웨어와 OPRoS 로봇 시뮬레이터는 해당 로봇 소프트웨어에서 사용하는 하드웨어를 제어하기 위한 Device API를 통해 연결된다. Device API는 OPRoS 로봇 시뮬레이터와 통신을 하며 OPRoS 로봇 시뮬레이터에 로드된 환경 모델(EML)로부터 로봇이 구동하는 환경 정보를 획득하고 이 환경모델 내의 로봇모델(AML)을 제어한다.

OPRoS 컴포넌트들을 기반으로 구현된 로봇 소프트웨어는 그림 6과 같이 서로 다른 또는 같은 종류의 OS의 여러 서버에 분산 배치할 수 있다. 이 때 해당 서버의 OPRoS 컴포넌

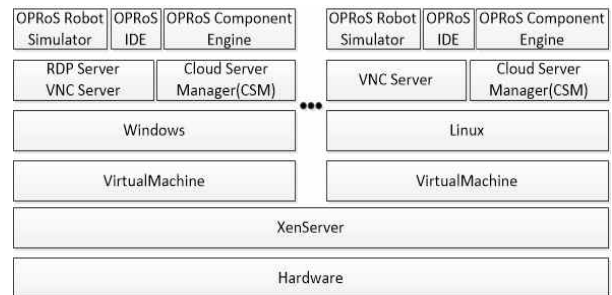


그림 4. 클라우드 기반 OPRoS용 통합개발환경.  
Fig. 4. Cloud-based IDE for OPRoS.

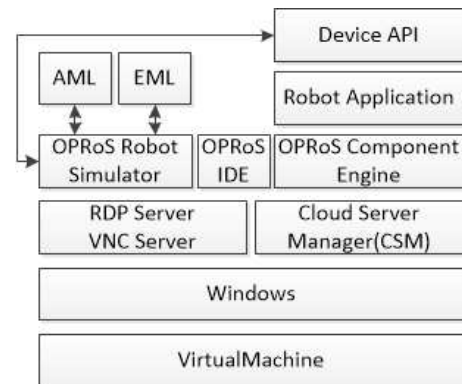


그림 5. 로봇 응용과 OPRoS 로봇 시뮬레이터간 관계.  
Fig. 5. Relationship between robot application and OPRoS robot simulator.

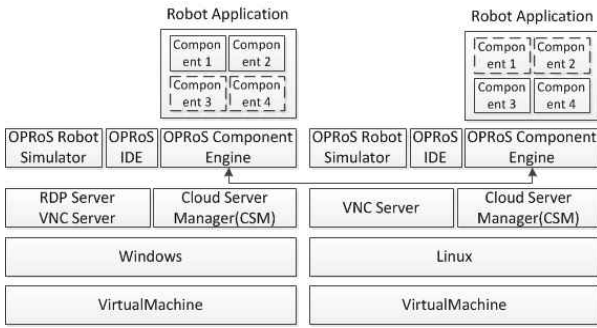


그림 6. OPRoS 응용의 분산 배치.  
Fig. 6. Distributed deployment of OPRoS application.

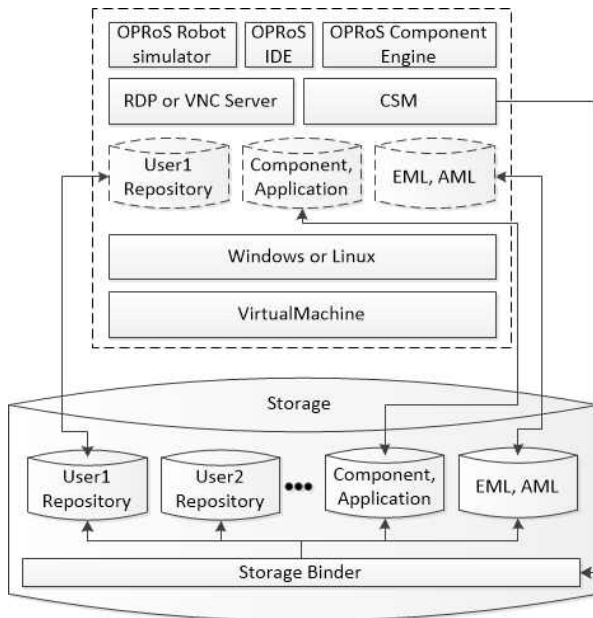


그림 7. 클라우드기반 OPRoS 플랫폼과 저장소간의 관계.  
Fig. 7. Relationship between the cloud-based OPRoS platform and the storage.

트 엔진끼리 통신하며 각 컴포넌트의 서비스를 호출하고 데이터를 전송한다. 그림 6의 컴포넌트 1, 컴포넌트 2, 컴포넌트 3, 컴포넌트 4로 구성된 로봇 소프트웨어는 Windows 서버에 컴포넌트 1과 컴포넌트 2를 배치되고 Linux 서버에 컴포넌트 3과 컴포넌트 4가 배치되어 동작한다.

OPRoS 플랫폼을 적용한 클라우드 플랫폼에서 사용하기 위한 저장소는 그림 7과 같이 구현되었다. 저장소는 크게 사용자의 개발 산출물을 저장하기 위한 사용자 개인 저장소와 개발자의 편리한 개발을 위해 플랫폼에서 제공하는 개발용 저장소로 나뉘어진다. 개발용 저장소에는 OPRoS IDE에서 활용하기 위한 OPRoS 컴포넌트와 이 컴포넌트들로 구현된 애플리케이션이 포함되고 OPRoS 로봇 시뮬레이터에서 활용하기 위한 환경모델과 로봇모델이 포함된다. OPRoS IDE와 OPRoS 로봇 시뮬레이터에서는 파일 단위로 개발 산출물을 생성하고 로드하기 때문에 모든 저장소는 파일 단위로 구축되어 있다. 각 저장소는 사용자가 서버가 연결될 때 공유폴더를 통해 해당 서버와 연결되고 사용자

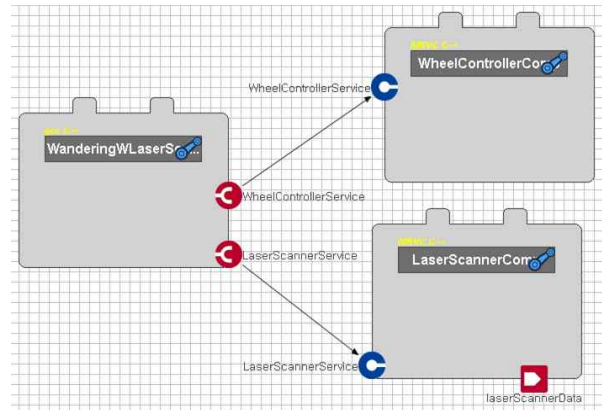


그림 8. 주행 응용 예.  
Fig. 8. An example for navigation application.

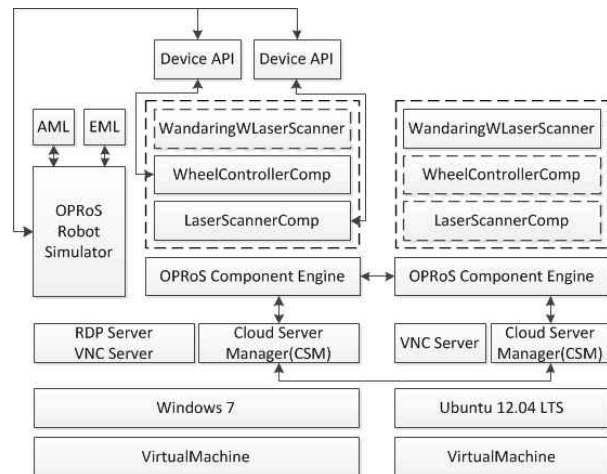


그림 9. 주행 응용 예제 시험을 위한 클라우드기반 OPRoS 플랫폼.  
Fig. 9. Cloud-based OPRoS platform for test of the navigation application example.

의 연결 해제 시 저장소의 연결도 해제된다. 사용자 개인 저장소는 읽기/쓰기가 모두 가능하기 때문에 코걸 저장소와 같이 자유롭게 사용할 수 있다. 개발용 저장소는 [10]에서 제공하는 OPRoS 표준 컴포넌트 및 애플리케이션과 시뮬레이션을 위한 로봇모델, 환경모델 등을 기본적으로 포함하며 사용자로부터 개발된 컴포넌트, 애플리케이션 및 로봇모델과 환경모델 또한 배포가 가능하여 다양한 로봇 및 환경을 공유할 수 있도록 한다.

구현된 플랫폼에서 그림 8과 같은 OPRoS 컴포넌트를 기반으로 한 주행 로봇 소프트웨어를 구현하고 시뮬레이션 하였다. 구현된 주행 응용은 WanderingW Laser Scanner, LaserScannerComp, 그리고 WheelControllerComp 3가지의 컴포넌트로 구성되어있다. LaserScannerComp는 레이저 스캐너 센서를 제어하고 WheelControllerComp는 로봇의 바퀴를 제어하며 WanderingW LaserScanner는 위의 두 컴포넌트를 제어해 로봇이 장애물을 회피하고 이동하도록 한다. 참고로 전자의 2개는 윈도우즈용이며 후자는 리눅스용이다.

랜덤 주행 로봇 소프트웨어는 그림 9와 같이 분산배치되

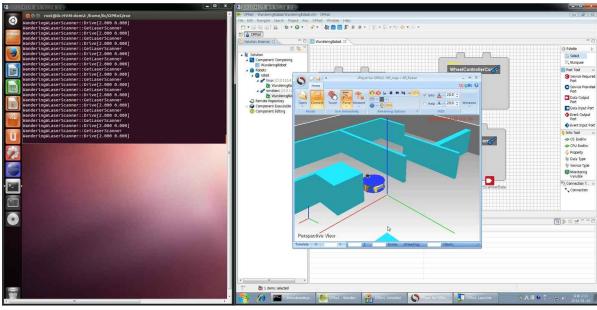


그림 10. 클라우드 기반 OPRoS 통합 개발 환경에서의 주행 응용 동작 영상.

Fig. 10. Moving picture of the navigation application example on Cloud-based OPRoS IDE.

어 동작한다. WheelControllerComp와 LaserScannerComp는 윈도우즈 서버에 배치되며 WanderingWLaserScanner는 리눅스 서버에 배치된다. 로봇의 바퀴와 레이저 스캐너 센서를 제어하는 WheelControllerComp와 LaserScannerComp는 Device API를 통해 OPRoS 로봇 시뮬레이터와 연동되어 동작하며 OPRoS 로봇 시뮬레이터에서 로봇의 동작을 확인할 수 있다. OPRoS 컴포넌트 엔진간의 통신을 위해서는 통신 대상 OPRoS 컴포넌트 엔진이 구동되는 서버의 IP와 OPRoS 컴포넌트 엔진의 포트가 필요하다. 이 두 정보를 OPRoS 컴포넌트 엔진의 시스템 환경 파일에 적용하여 동작시키기 되된다. 제안하는 시스템에서는 이를 CSM에서 사용자가 접속한 서버들의 정보를 이용해 자동으로 설정하여 동일한 OS 또는 서로 다른 OS의 여러 서버에서 구동되는 OPRoS 로봇 응용을 쉽게 구동할 수 있다.

그림 9의 로봇 소프트웨어가 구현된 전체 클라우드 환경에서 동작하는 화면은 그림 10에 나타나 있으며 [11]에서 동작 영상을 확인할 수 있다.

#### IV. 결론

본 논문에서는 HW를 시험할 수 있도록 로봇 시뮬레이션을 제공하고 다양한 운영체제와 다중 보드를 가지는 로봇을 위하여 로봇 소프트웨어를 쉽게 개발하고 시험할 수 있는 클라우드 기반 통합개발환경을 제안하였고 OPRoS를 사용하여 제안된 시스템의 유용성을 검증하였다.

제안하는 클라우드 기반 통합개발환경은 클라우드의 확장 가능한 인프라스트럭처를 기반으로, 1개 이상의 운영체제를 가진 다중 보드 기반 로봇을 위한 소프트웨어의 개발 및 테스트를 가능하도록 제안하였다. 또한 사용자 개인 저장소 및 개발용 저장소를 포함하여 개발 및 테스트를 위한 다양한 자원을 제공하고 여러 로봇 소프트웨어 개발자의 개발 산출물을 배포하고 공유하는 것이 가능하도록 하였다. 향후 로봇 소프트웨어의 시뮬레이션뿐만 아니라 실제 장치를 통한 동작을 지원하기 위한 실제 HW 장치가 포함된 클라우드 환경에 대한 연구가 진행되어야 할 것이다.

#### REFERENCES

[1] Cloud, [http://en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing)

[2] J. Terazono, H. Fukuhara, T. Yamada, T. Nihei, K. Suzuki, I. Kosedo, R. Fujita, T. Miyazaki, S. Saito, A. Kara, and T. Hayashi "A sensor network using content-aware messaging network architecture," *ICCAS-SICE International Joint Conference*, pp. 5055-5058, Aug. 2009.

[3] Y. Chen, Z. Du, and M. Garcia-Acosta, "Robot as a service in cloud computing," *Proc. of IEEE International Symposium on Fifth Service Oriented System Engineering*, pp. 151-158, 2010.

[4] R. Arumugam, V. R. Enti, L. Bingbing, W. Xiaojun, K. Baskaran, F. F. Kong, A. S. Kumar, K. D. Meng, and G. W. Kit, "DAVINci - a cloud computing framework for service robots," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 3084-3089, May 2010.

[5] Z. Du, W. Yang, Y. Chen, X. Sun, X. Wang, and C. Xu, "Design of a robot cloud center," *Proc. of International Symposium on Autonomous Decentralized Systems*, pp. 269-275, March 2011.

[6] K. Kamei, S. Nishio, N. Hagita, and M. Sato, "Cloud networked robotics," *IEEE Network Magazine*, vol. 26, no. 3, pp. 28-34, May-Jun. 2012.

[7] T. Aho, A. Ashraf, M. Englund, J. Katajamäki, J. Koskinen, J. Lautamäki, A. Nieminen, I. Porres, and I. Turunen, "Designing IDE as a service," *Communications of Cloud Software*, vol. 1, no. 1, Dec. 2011.

[8] T. Mikkonen and A. Nieminen, "Elements for a cloud-based development environment: online collaboration, revision control, and continuous integration," *Nordic Symposium on Cloud Computing and Internet Technologies*, pp. 14-20, Aug. 2012.

[9] L. M. Gadhikar, L. Mohanv, M. Chaudhari, P. Sawant, and Y. Bhusara, "Browser based IDE to code in the cloud," *Advances in Intelligent Systems and Computing*, vol. 203, pp. 59-69, 2013.

[10] OPRoS, <http://www.ropros.org>

[11] <http://www.youtube.com/watch?v=EK07fQgwDFQ>

[12] S. Han, M. Kim, and H. S. Park, "Open software platform for robotic services," *IEEE Transactions on Automation Science and Engineering (in Korean)*, vol. 9, no. 3, pp. 467-481, 2012.

[13] S. Kim and H. S. Park, "Design of a robot-in-the loop simulation based on OPRoS," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 19, no. 3, pp. 248-255, 2013.

[14] J. Y. Jang, S. H. Ji H. S. Park, "Geographical Group-based FastSLAM Algorithm for Maintenance of the Diversity of Particles," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 19, no. 10, pp. 907-914, 2013.

**윤재훈**

2012년 강원대학교 전자통신공학과 학사 졸업. 2014년 동 대학원 석사 졸업. 관심분야는 소프트웨어 공학, 소프트웨어 테스트 자동화, 클라우드 컴퓨팅.

**박홍성**

1983년 서울대학교 제어계측공학과 학사 졸업. 1986년 동 대학원 석사. 1992년 동 대학원 박사. 1992년~현재 강원대학교 전자통신공학과 교수. 관심분야는 로봇 S/W 플랫폼, 무선데이터통신, 실시간 통신.