
4족 보행로봇 및 MSRS를 이용한 분산 웹 개발환경 구축

양태규* · 서용호**

Construction of Distributed Web Development Environment using 4-Legged Robot and MSRS

Tae-Kyu Yang* · Yong-Ho Seo**

본 연구는 2008년도 산학협동재단 학술연구비 지원사업의 지원을 받아 수행되었음

요 약

본 논문은 4족 보행로봇 및 MSRS를 이용한 분산 웹 환경 로봇 서비스 개발환경을 제안하였다. 로봇 서비스 개발 환경은 고수준의 복잡한 로봇어플리케이션이 분산 웹 환경에서 동작 가능하게 하기 위하여, 마이크로소프트사의 MSRS를 이용해 로봇 기능들을 네트워크 노드에서 실행되는 서비스들로 구현하였다. 제안된 개발환경은 4족 보행 로봇인 제니보를 로봇플랫폼으로 선정하여 구현되어 졌다. 실험을 통해 개발된 로봇 서비스들은 분산 웹 환경 하에서 동적인 실행, 구성 및 원격 모니터링이 가능함을 보여주었다. 또한 개발된 로봇 서비스들을 복잡한 로봇어플리케이션을 쉽게 제작할 수 있도록 하는 비주얼프로그래밍언어에서 사용하여 응용어플리케이션을 제작해 그 가능성을 검증하였다.

ABSTRACT

This paper proposed a distributed web environment for developing robot services using a 4-legged robot and MSRS. The proposed development environment of the robot services enabled a high-level complicated robot application to operate in a distributed web environment by implementing robot functions as services which can be run on network nodes using Microsoft MSRS. The environment has been also implemented using a 4-legged robot, Genibo as a robot platform. In experiment, we showed the developed robot services are enable to be executed and configured dynamically and support a remote monitoring in a distributed web environment. Finally the feasibility and the effectiveness of the proposed environment has been verified by creating a robot application using the developed robot services in Visual Program Language which helps student to create a complicated robot application system easily.

키워드

4-Legged Robot, Distributed Web Environment, Robot Service Programming, High-level Robot Task

* 목원대학교 지능로봇공학과

** 남서울대학교 컴퓨터학과

접수일자 2009. 04. 01

심사완료일자 2009. 05. 27

I. 서 론

지능로봇 기술은 다양한 하드웨어와 소프트웨어가 접목되어 있는 복합적 요소기술을 필요로 한다. 이를 위해서는 기존의 단순 개발환경과는 다른 새로운 패러다임을 가지는 개발환경이 요구되고 있다. 이러한 지능로봇을 위한 새로운 개발환경은 로봇 내부의 하드웨어 디바이스들을 단순화 구조화시켜 이식성 및 확장성을 보장할 수 있어야 한다. 그리고 이러한 추상화된 디바이스들을 이용하여 로봇 행위를 체계적으로 제어할 수 있으며, 특히 네트워크 환경에서 각각의 지능로봇 기술들이 운용될 수 있는 분산 컴포넌트 환경을 고려한 아키텍처의 설계가 필요하다.

기존의 로봇을 위한 소프트웨어 개발환경 및 로봇 운용을 위한 미들웨어환경들은 크게 디바이스 간의 메커니즘 측면, 컴포넌트 모델 측면, 아키텍처 측면으로 나누어 다음과 같은 기술들이 제안되어졌다[1].

첫째, 디바이스 간의 메커니즘을 제공한 **MIRO (Middleware for ROBot)**의 경우 디바이스들을 체계적으로 추상화하고 **CORBA**를 적용하여 이들을 외부에서 제어한다. **The Player & Stage**는 소켓통신을 이용해 로봇의 디바이스들을 외부에 노출하여 멀티로봇시스템을 위한 프레임워크를 제시하고 있다. 이와 같은 하드웨어 추상화는 로봇 응용의 재사용성과 유지 보수를 크게 향상시켰다[2-3].

둘째, 컴포넌트 모델 측면에서 **EURON (European Robotics Research Network)**의 지원 하에 유럽 4개의 연구기관이 참여한 특정 플랫폼이나 특정 로봇 응용에 종속적이지 않는 일반적인 로봇제어를 위한 기본 기능을 제공하기 위한 연구로 로봇제어에 관한 코어라이브러리와 이벤트 처리, 컴포넌트 라이프 사이클 관리를 목적의 **Orocos (Open ROBot COnTrol System)**와 일본 산업기술총합연구소(AIST) 산하 지능시스템 연구부문에 로봇을 구성하는 요소 기술들을 통신 네트워크를 통해 조합함으로써, 다양한 로봇을 빠르고 손쉽게 구축할 수 있도록 로봇의 기반 미들웨어 기술을 확립하고 표준화를 목표로 하여 개발 중인 **RT (Robot Technology)** 미들웨어가 있다[4-5].

셋째, 아키텍처 측면에서 살펴볼 때 **ERSP (Evolution Robotics Software Platform)**는 Evolution Robotics사가 2002년 개발한 로봇 소프트웨어 플랫폼 **ERSP**는 기본

적으로 모바일 로봇의 제반 소프트웨어를 개발하기 위한 플랫폼으로서, 행동기반 로보틱스의 개념을 지원하며, 계층구조를 통한 모듈화, 서로 다른 운영체제 및 하드웨어를 지원하기 위한 이식성, 커스터마이징, 하드웨어 독립성, 그리고 새로운 하드웨어의 추가가 용이한 확장성을 제공하기 위해 **태스크 실행 계층 (Task Execution Layer-TEL)**, **행위 실행 계층 (Behavior Execution Layer- BEL)**, 그리고 하드웨어 추상화 계층 (**Hardware Abstraction Layer-HAL**)의 세 계층으로 구성된다[6].

이러한 개발환경 중 **Orocos**의 컨트롤 커널, **RT** 미들웨어의 컴포넌트 모델 등이 대표적인 데이터 통신 포트에 기반한 제어 구조를 가지고 있다. 특히 **RT** 미들웨어 컴포넌트 모델의 경우 내부적으로 스레드를 구동할 수 있는 모델을 가지고 있어서, 주기적 작업을 처리하거나 비동기적 통신이 필요한 다양한 로봇 응용에 적합한 구조라 할 수 있다. 하지만 이들은 로봇 컴포넌트 간의 실시간 통신과 메시지 버퍼링 등과 문제들은 아직 고려하지 못하고 있거나 현재 개발 중에 있으며, 웹 환경에서의 여러 로봇 기술이 분산되어 작동하는 로봇을 고려하지 않고 있다.

따라서 본 논문에서는 이러한 분산 웹 환경에서 로봇 소프트웨어를 개발하고 운용하기 위해 최근 제안된 마이크로소프트의 **MSRS (Microsoft Robotics Studio)**를 이용하여 4족 보행로봇을 분산 웹 환경에서 동작 가능한 개발환경을 구축하고 이의 효용성을 검증하였다[7-8].

특히 개발된 분산 웹 개발환경은 해당 로봇을 위한 다양한 로봇기술들이 컴포넌트화 되어 로봇과 웹 환경에 분산된 여러 컴퓨터들에서 유기적으로 연동이 가능하도록 시스템을 설계하고 구현하였다. 마지막으로 실험을 통해 개발된 로봇 서비스들이 분산 웹 환경 하에서 원격 모니터링 및 서비스의 동적인 실행 및 구성이 가능함을 보여주었으며, 또한 개발된 로봇 서비스들을 이용해 복잡한 로봇어플리케이션을 쉽게 제작할 수 있도록 하는 비주얼프로그래밍언어에서 사용하여 4족 로봇용 어플리케이션을 제작해 그 가능성을 검증하였다.

II. 4족 보행로봇 제니보

본 논문에서 구현하고자 하는 분산 웹 로봇 개발환경은 로봇 기술이 단순히 한 개의 프로세서에서 실행되는 로봇이 아닌 네트워크에 분산된 다양한 로봇 서비스들이 결합된 구조를 통해, 보다 확장가능하고 복잡한 구성을 지닌 로봇 운용환경이다. 따라서 이러한 고수준의 로봇 개발환경을 잘 보여주기 위해 적합한 로봇으로, 단순한 바퀴 구조가 아닌 다관절의 복잡한 보행구조를 가지고 다수의 이중 입출력이 가능한 4족형 보행로봇을 본 연구용 로봇 플랫폼으로 선정하게 되었다.

4족 보행로봇은 수십 개의 모터와 이를 제어할 수 있는 제어장치, 카메라를 비롯해 센서, 기구 가공물 등 수백 가지의 부품과 보행 알고리즘, 영상처리 알고리즘, 음성인식 알고리즘 등 다양한 하드웨어 및 소프트웨어 기술의 복합되어 있는 로봇이다. 따라서 로봇전용 SW 개발환경을 이용한 4족 로봇 적용 예는 찾아보기 힘들며, 제공하는 로봇 응용기술들이 단순하고 특히 보행제어 소프트웨어 중심의 개발만이 이루어져 왔다.

4족 로봇들 중 대표적인 소니의 AIBO 시리즈는 상업적으로 가장 성공한 4족 보행로봇이다. AIBO의 경우 다양한 음성인식 및 영상처리를 수행할 수 있어 인간 로봇 상호작용 기술을 적용할 수 있으며 정적 보행이 가능하다. 하지만 소니의 AIBO는 제품생산이 중단되어 현재 시중에서 쉽게 구할 수 없다[9].

따라서 본 논문에서는 기존 출시된 4족 보행 로봇들과 각 로봇 별 API 제공 여부를 조사한 결과, 로봇의 센서 입력 및 표현기능의 종류가 많고 고수준의 로봇 어플리케이션 개발에 적합한 다사로봇의 제니보(Genibo)를 최종 로봇플랫폼으로 선택하였다[10].

2.1 제니보 센서 및 하드웨어 구성

제니보는 그림 1과 같이 각 부분별로 다수의 다양한 종류의 센서를 가지고 있다. 또한 제니보M은 표 1과 같이 마이크, 기울기 센서, 음성인식, 15개의 관절(Joint) 모터와 3Color Dot matrix 등 15가지의 입출력 사양을 갖는 4족 보행로봇이며, 로봇 전체의 하드웨어 사양은 표 2와 같다.

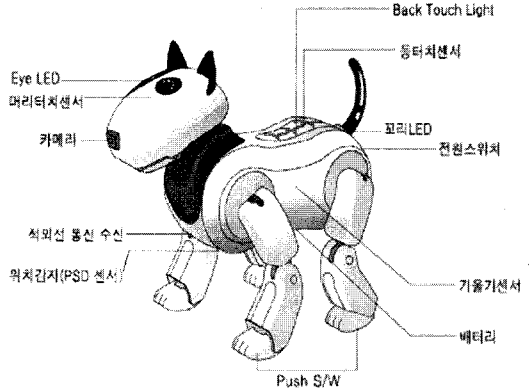


그림 1. 제니보 센서 구성도
Fig. 1 Genibo Sensor Structure

표 1. 제니보 입출력 사양
Table. 1 Genibo Input/Output Specification

입력	출력
카메라	Color : 3(Blue, Red, B+R)
Tilt 센서	3Color Dot Matrix
거리측정모듈	다리 12개 모터
터치감지센서	머리 2개 모터
음성인식	꼬리 1개 모터
바닥 접촉감지	스피커

표 2. 제니보의 하드웨어 사양
Table. 2 Genibo Hardware Specification

사양	가로 x 세로 x 높이	328 x 279 x 193mm(L x H x W)
무게	배터리 포함	1.5kg
메인보드	32bit RISC Processor	연산속도 : 350MHz FlashRom 32Mbytes SDRAM 64Mbytes
카메라	1.3M pixel, 1/4inch CMOS	Image size : 1280 x 1024 Frame Range : 15fps 이상 Image format : JPEG
센서	RTC 모듈	내장
	Tilt 센서	내장
	거리측정모듈	인식거리 10~80cm
	터치감지센서	머리, 등(4), 꼬리(좌,우)
	음원위치 추적센서	마이크 3개
음성인식	마이크 1개	
감정표현	LED보드(이모티콘)	Color : 3(Blue, Red, Blue+Red) 3Color Dot Matrix : 31LEDs(눈 안쪽당)
전원관리	배터리	전압 : 7.4V, 2200mAh
	배터리충전모듈	내장
어댑터		입력전원 : AC 110/220 at 50Hz~60Hz 자동 전환 출력전원 : DC 9V, 2.5A

본 논문에 사용된 4축 보행로봇 제니보는 개발자용 버전으로 출시된 제니보M으로 컴퓨터와의 무선 네트워크 통신을 기반으로 15축의 모토, 기울기 센서, 적외선 통신, 터치센서 등이 장착되어 있어, 다양한 응용환경을 제공하고 있으며, MSRS에서 제공하는 서비스들과 결합하여 전자, 기계, 소프트웨어, 정보통신 등 로봇과 관련된 다양한 교육 분야에서 활용이 가능하다.

2.2 제니보 소프트웨어 API

제니보M은 TCP/IP기반 소켓 통신을 사용해 로봇 각 세부 기능을 제어하는 API(Application Programming Interface)를 지원하고 있다. 윈도우 DLL기반의 라이브러리를 통해 Microsoft Visual C++, Visual Basic과 Borland C++, Delphi등 다양한 프로그램 개발환경 지원하고 있다.

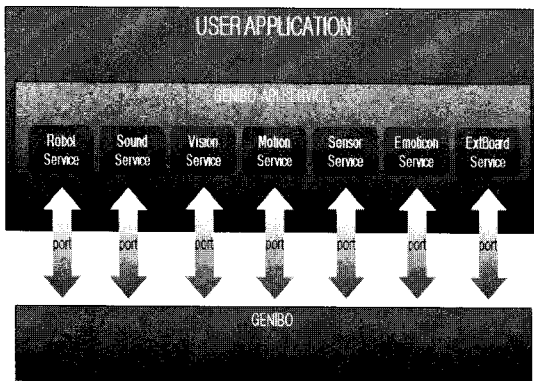


그림 2. 제니보 소프트웨어 API구조
Fig. 2 Genibo Software Library API Structure

표 3. 제니보의 API 사양
Table. 3 Genibo API Specification

로봇기능	로봇기능 상세설명
Robot	로봇 설정에 대한 정보 서비스
Sound	사운드 재생, 녹음, 음성 인식 서비스
Vision	영상 녹화, 관리 서비스
Motion	모션 재생, 관리 서비스
Sensor	센서 정보 서비스
Emoticon	이모티콘 재생, 관리 서비스
ExtBoard	확장 RS232보드 관리 서비스

제니보M에서 제공되는 API는 그림 2와 같이 각 로봇 기능별로 별도의 소켓 포트가 할당된 구조로 다음의 표 3과 같은 기능들을 사용할 수 있다.

III. 분산 네트워크 프로그래밍을 위한 MSRS

MSRS는 마이크로소프트에서 출시한 로봇 개발자가 손쉽게 로봇 애플리케이션을 개발하기 위한 윈도우 기반 개발환경이다. MSRS는 서비스중심으로 개발이 이루어지도록 하며, 시뮬레이션을 통해 개발된 서비스를 실제 로봇에 쉽고 빠르게 적용할 수 있는 개발환경이다. 이 개발 틀은 특히 C++, C#등의 기존 프로그래밍 언어 뿐만 아니라, 다이어그램 방식으로 데이터와 서비스를 조합하여 표현하는 비주얼프로그래밍언어(VPL)를 내장하고 있다. 또한 MSRS는 .NET framework을 사용하는 다양한 종류의 윈도우 플랫폼과 .NET Compact Framework를 사용하는 Windows CE를 지원하여 임베디드 시스템에서의 동작이 가능하다[8].

개발된 서비스를 실행해주는 런타임은 PC에 직접 연결된 (시리얼 포트, 블루투스, USB등) 로봇으로부터 온 보드 PC를 가진 로봇, 시뮬레이션환경 하에서 작동하는 로봇까지 다양한 로봇을 지원한다. 이 런타임은 기본적인 센서 입력의 관측(Observation)에서 연결에 의한 구동(Drive-by-wire) 과 원격 작업(Remote presence) 및 자동화된 업무, 다수의 자동화된 로봇들의 협력 등의 다양한 종류의 로봇어플리케이션을 지원한다.

MSRS의 하부구조인 런타임은 다음의 두 주된 요소인 CCR과 DSS를 통해 구성되며 이를 통해 분산 웹 환경에서 로봇 서비스를 지원한다.

3.1 동시처리 및 제어기술(CCR)

동시처리 및 제어기술인 CCR(Concurrency and Coordination Runtime)은 기존의 다중 작업(multitasking) 처리를 위한 멀티 스레드(Multi-Thread) 프로그래밍 방식을 대폭 개선한 기술로 상당히 복잡한 멀티스레드 프로그래밍 기술이 요구되는 스레드간의 데이터 통신 등의 작업을 손쉽게 구현할 수 있어 로봇의 영상, 센서 등의 다양한 입·출력을 동시에 처리할 수 있도록 한다 [12].

설계단계에서부터 네트워크 애플리케이션을 제작 방법과 개념을 이 기종 하드웨어의 통합에 도입하여 손쉽게 동시성, 디버깅, 고립화(Isolation)를 일관되게 처리할 수 있게 함으로써 컴포넌트들을 독립적으로 개발될 수 있게 하였다. 이를 통해 CCR은 비동기 오퍼레이션(Asynchronous operations) 관리와 동시성 처리 및 병렬처리 하드웨어, 부분적 실패(Partial failure) 처리를 손쉽게 하는 서비스를 제공한다. 이러한 기능은 특정 로봇에 종속되지 않은 로봇의 구성요소 각각을 디자인할 수 있도록 하여 컴포넌트 측면에서의 구현된 기술임을 나타낸다.

3.2 분산화된 소프트웨어 서비스(DSS)

웹 아키텍처의 장점은 단순성, 상호호환성, 느슨한 연결(Loose coupling)이다. 특히 HTTP는 단순하고, 상태(State)중심의 애플리케이션 모델인 “REST (Representational State Transfer)”의 장점을 지닌 애플리케이션을 가능하게 하는데 기여하여 왔다. 웹기반 애플리케이션은 HTTP를 통해 확장가능하며 상호운용성과 다양한 시나리오를 만족하는 유연성을 보여 왔다. 분산된 지능로봇 애플리케이션을 웹을 통해 접근, 수집하여 재조립 및 확장이 가능하도록 하기위해 MSRS는 이 웹서비스로 대표되는 REST 기반의 일정한 형식을 제공하고 있다[11-12].

기존의 웹기반 아키텍처인 REST(Representation State Transfer)모델을 기반으로 SOAP(Simple Object Access Protocol) 방식과 HTTP방식의 장점만을 모아 서비스 기반 프로그래밍에 적합하도록 개발한 인터페이스 기술이 분산화된 소프트웨어 서비스인 DSS(Decentralized System Services)이다[12].

DSS는 서비스들 간의 상호작용을 위해 SOAP과 HTTP를 연동한 DSSP(Decentralized System Service Protocol) 기본으로 구성되어 졌다. 이를 통해 분산 웹 환경에서의 서비스조작, 이벤트 처리, 메시지 교환이 가능하다. 또한 DSS는 서비스 호스팅 환경을 제공하여 디버깅, 기록, 감시, 보안 등과 같은 제공한다. 이러한 서비스 구조는 또한 디바이스를 네트워크 및 웹에 노출 시켜 디바이스의 재사용성과 유지보수를 위한 웹기반의 디바이스 간 인터페이스 메커니즘을 제공해 준다.

DSS는 기존의 REST 인프라구조와 상호운용성을 잃지 않으면서 이벤트 알림 모델과 구조화된 데이터 조

작의 장점을 가질 수 있게 되었으며, 그 결과로 노드(Node) 내 혹은 네트워크상에서 분산된 서비스들로 로봇 기술들을 구성해 상호작용적이고 동적인 분산 웹 환경 애플리케이션이 가능하게 되었다. 이러한 DSS를 이용해 로봇에서 발생하는 모든 이벤트를 서비스의 상태 변화로 모델링하여 로봇서비스를 REST모델로 다루게 되었다.

3.3 CCR과 DSS를 이용한 분산 웹 환경 로봇 서비스 개발

MSRS는 로봇의 여러 센서에서 들어오는 입력들과 로봇의 모터를 포함한 다양한 출력들에 대한 비동기 연산(Asynchronous operations)들로 정의한다. 또한 이들의 비동기 연산의 집합을 분산 웹 환경에서 지원하기 위해 CCR과 DSS를 이용한다.

CCR은 다중 프로세서 하드웨어 구조를 지원하면서 부분적 실패(Partial failure handling)를 처리하기 위한 동시성 메시지 기반 프로그래밍 모델(Concurrent message-oriented programming model)로 로봇의 여러 기능들을 동시에 제어하기가 기존 프로그램 환경에 비해 훨씬 용이하다.

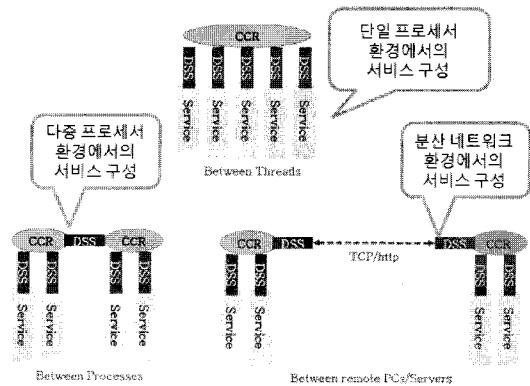


그림 3. CCR과 DSS를 이용한 로봇 서비스 구성
Fig. 3 Robot Service Coordination using CCR & DSS

원격 서비스나 제어 등에 필요한 통신 환경을 위한 DSS는 멀티스레드, 멀티 프로세스, 원격 통신 프로그래밍 등에 CCR과 연동하여 하나의 PC에서 개발하는 것과 동일한 방식으로 분산 웹 환경에서도 일관된 프로그래밍 방법을 사용하게 해 주어, 로봇 애플리케이션을 보다

용이하게 개발하게 해 준다. 그림 3은 CCR과 DSS를 연동한 분산 웹 환경까지 확장 가능한 멀티스레드, 멀티프로세스, 원격 서비스 지원구조를 보여준다.

IV. 4족 로봇을 위한 분산 네트워크 프로그래밍 환경 개발

4.1 서비스 구성요소(Service Components)

서비스는 MSRS 애플리케이션을 제작하기 위한 기본적인 단위이며, DSS 애플리케이션의 모델의 주요 구성요소이다. 로봇을 위한 서비스들은 크게 다음 세 가지 기능을 포함한 표현에 사용된다. 첫째는 센서와 액츄에이터와 같은 하드웨어 기본 구성 서비스, 둘째는 유저 인터페이스, 기억 장치, 디렉터리 서비스, 등과 같은 소프트웨어 구성 요소로서의 서비스, 마지막으로 센서 융합, 메쉬업(Mash-ups) 등과 같은 서비스의 집합체로서 서비스와 로봇의 기본 서비스를 관리하는 코디네이션 기능의 서비스이다.

한 서비스는 DSS노드 안에서 실행된다. DSS노드는 서비스를 생성 및 관리해주는 호스팅 환경이다. 서비스는 기본적으로 네트워크 기능을 가지고 있다. 따라서 서비스는 한 개의 DSS 노드 내에서 뿐만 아니라 네트워크 상의 다른 DSS노드에 이르기 까지 동일한 방법으로 서로 상호작용할 수 있다.

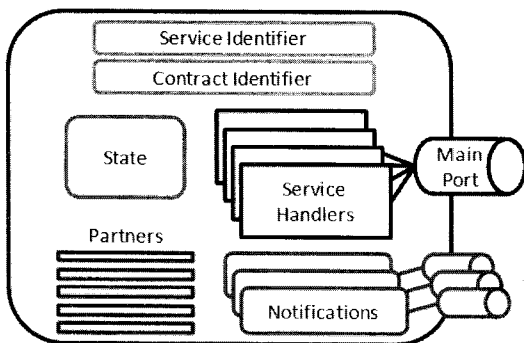


그림 4. 서비스 구성요소
Fig. 4 Service Components

DSS 서비스 모델은 서비스의 재사용이 용의하고, 서비스들 사이의 느슨한 연결 구조를 이루게 되어 서로 조

립하고 사용하기 용이하게 디자인되어 있다. 모든 DSS 서비스는 그림 4와 같은 서비스 컴포넌트의 공통집합으로 구성된다.

4.2 서비스 핸들러(Service handlers)

각 서비스는 메인포트를 통한 메시지 통신을 통해 상호작용한다. 다른 서비스에서 전달된 메시지는 해당 서비스의 포트에 입력되며, 서비스 핸들러를 통해 메시지들을 처리한다. 예를 들어 기본 웹 서비스의 경우 기본 서비스 상태 조작을 위한 Get, Replace에 대한 핸들러 이외에 HttpGet와 HttpPost를 위한 핸들러를 등록해 처리할 수 있다.

서비스 핸들러 내에서 해당 서비스는 또한 다른 서비스로 메시지에 보낼 수 있다. 이때 메시지를 보내는 두 가지 방법으로는 다른 서비스에 보내는 요청 메시지와 서비스 내에서 상태 변경의 결과를 서브스크라이버(Subscriber) 서비스를 통해 보내는 이벤트 알림 방식이 있다. 두 가지 방법 모두에서 전달하고자 하는 메시지는 상대 서비스의 메인 포트를 나타내는 CCR포트인 서비스 포워더(service forwarder)를 통해 그림 5와 같이 전해진다. 메시지가 서비스 포워더를 통하여 보내질 때, MSRS의 하부구조인 런타임은 네트워크상으로 메시지를 다른 서비스로 전달하는 트랜스포트에 도달할 때까지 연속해서 전달하며, 수신측 서비스의 메인 포트에 도착할 때까지 유지된다.

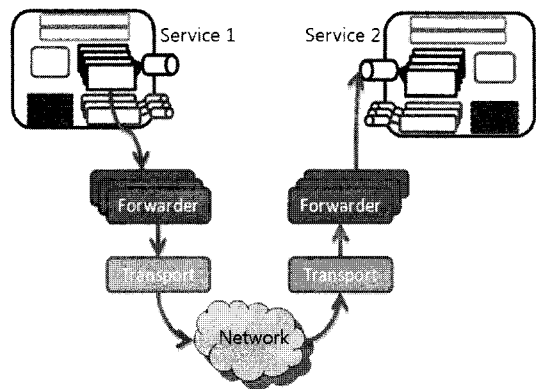


그림 5. 서비스 포워더를 통한 메시지 전달
Fig. 5 Message Transmission using Service Forwarder

4.3.4족 로봇 제니보 서비스 개발

MSRS는 로봇 소프트웨어를 용이하게 개발하기 위해 로봇에 사용되는 기능과 역할을 위한 분류하고, 각 로봇 디바이스별 입출력 사양을 기준으로 이를 표준화하여 기본 서비스를 제공하고 있다. 그림 6은 MSRS에서 정의된 기본 로봇 서비스를 포함한 전체 서비스 체계를 보여준다.

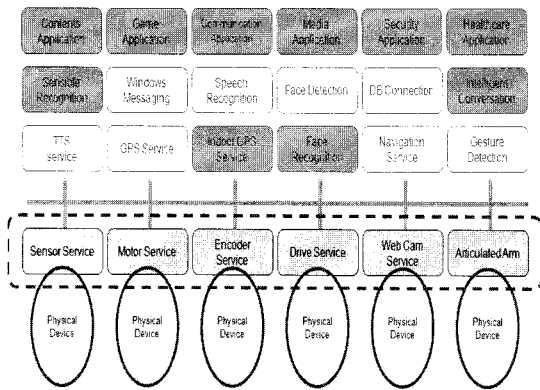


그림 6. MSRS 로봇 서비스 체계
Fig. 6 MSRS Robot Service Organization

센서서비스의 경우, 범퍼, 터치 등 접촉을 통해 정보를 획득하는 서비스로 그 내부 상태로 하드웨어 식별자, 사용자에게 의해 정해지는 센서의 이름, 감지된 상태, 센서의 위치 및 센서의 감지 시간을 제공한다. 이러한 기능의 조합은 센서의 현재 상태를 확인하여 로봇에 적용하기에 적당하다.

4족형 로봇 제니보의 소프트웨어 개발용 API 구성 요소의 MSRS에서 제공하는 기본 서비스 적용가능여부를 판단하여 이를 기준으로 4족 로봇의 서비스들을 다음과 같이 디자인하였다. MSRS의 기본 서비스에서 제공하는 서비스는 각 서비스 별로 제니보 API를 이용해 해당 서비스별 기능들을 구현하였으며, 여기에 포함되지 않는 서비스들은 자체 서비스를 새로 정의하여 구현하였다. 본 논문에서 개발된 제니보 서비스들은 표 4와 같다.

표 4. 4족 로봇 제니보 서비스
Table. 4 4-legged Robot Genibo Services

4족 로봇기능	로봇 서비스 종류	기능
카메라	Microsoft.robotics.sevices.WebCam	영상입력
Tilt 센서	Microsoft.robotics.sevices.AnalogSensor	기울기 측정
거리측정모듈	Microsoft.Robotics.sevices.SonarSensor	전방거리 측정
터치감지센서	Microsoft.Robotics.sevices.ContactSensor	인체 접촉감지
Color LED	EmoticonMotionSound (제니보용)	디스플레이
3Color Dot Matrix	EmoticonMotionSound (제니보용)	디스플레이
다리 12개 모터	Microsoft.Robotics.sevices.Articulated Arm	각 다리별 3축 모션제어
머리 2개 모터	PanTilt (제니보용)	머리 모션제어
스피커	EmoticonMotionSound (제니보용)	사운드
모션	EmoticonMotionSound (제니보용)	몸동작
보행	Microsoft.Robotics.sevices.Drive	이동
전원	Microsoft.Robotics.sevices.Battery	전원관리
로봇연동	GeniboRobot (제니보용)	API관리

V. 실험

본 논문에서는 구축된 로봇 서비스 개발환경의 분산 웹 환경에서 동작을 검증하기 하고 그 효율성을 보여주기 위해, 우선 개발된 제니보 웹캠 서비스를 이용한 원격 모니터링을 시도하였다.

MSRS의 실행플랫폼인 런 타임은 노드라 불리는 형태로 관리되며, 하나의 노드에 하나의 서비스 또는 여러 개의 서비스를 포함시킬 수 있고, 또한 여러 개의 노드를 구성하여 동시에 실행시킬 수 있다. 이러한 분산 실행환경을 처리하기 위해, DSS기술을 적용하였으며, 인터넷 탐색기(Internet Explorer)를 통해 이러한 분산 호스팅 상태를 모니터링 하고 제어할 수 있다.

그림 7은 개발된 제니보 서비스들이 단일 노드의 런 타임에서 동작되는 것을 보여준다. 각각의 서비스들은 또한 웹 환경에 분산된 어떤 노드에서도 실행이 가능하다.

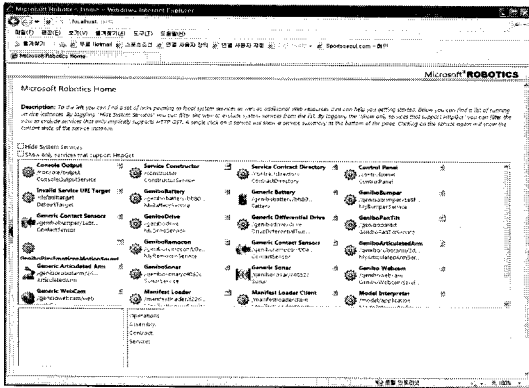


그림 7. 현재 PC노드에서 실행된 제니보 서비스를 제어하는 화면
 Fig. 7 Screen shot of Genibo Service Control in a Current PC Node

개발된 제니보 웹캠 서비스를 이용한 원격 모니터링은 웹캠 서비스의 내부 상태에 정의된 현재 로봇 카메라에서 저장된 이미지를 인터넷 탐색기를 통해 HTTP 서비스 요청하고, 그에 대한 응답으로 이미지를 획득하여 연속적으로 인터넷 탐색기 화면에 보여주는 방법으로 그림 8과 같이 동작이 이루어진다.

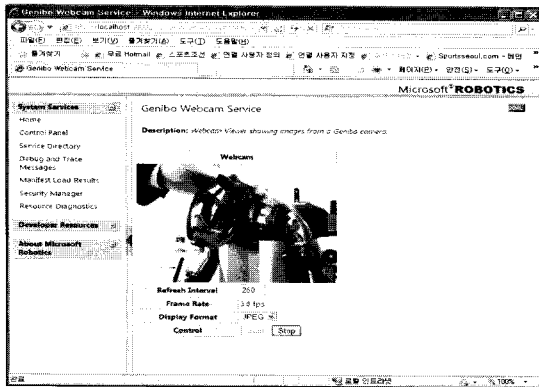


그림 8. 제니보 웹캠 서비스를 이용한 원격 모니터링 화면
 Fig. 8 Screen shot of Remote Monitoring using Genibo WebCam Service

본 논문에서는 또한 분산 웹 환경 로봇 어플리케이션을 쉽게 제작할 수 있도록 4족 로봇인 제니보의 구성, 기능, 성능을 분석하여 로봇 서비스들을 개발하였으며, 이를 MSRS에서 제공하는 비주얼프로그래머

(VPL)에서 손쉽게 재구성해 새로운 응용어플리케이션을 작성해 이를 실제 제니보 로봇에 적용하는 실험을 수행하였다. 그림 9는 제니보를 위한 VPL기반 개발 환경과 이를 이용해 제작된 응용어플리케이션을 보여준다.

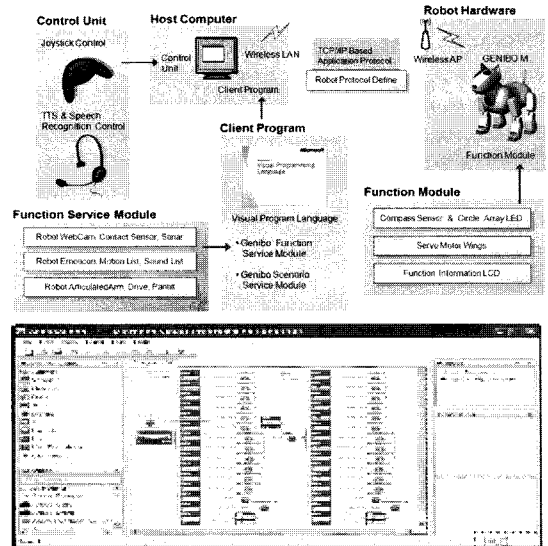


그림 9. 제니보를 위한 VPL기반 개발환경과 응용어플리케이션
 Fig. 9 Development Environment based on VPL for Genibo and its Application Examples

실험을 15자유도와 다양한 입출력, 음성인식, 영상처리 등 4족형 로봇 제니보의 모든 기능들을 구현된 제니보 서비스를 이용해 비주얼프로그래머로 개발된 응용어플리케이션에서 기존 프로그래머인 C++, C# 과 동일하게 사용할 수 있음을 보여주었으며, 이를 통해 구축된 로봇 서비스 개발환경의 신뢰성과 가능성을 검증할 수 있었다.

VI. 결론

본 논문에서는 복잡한 로봇 어플리케이션 개발을 위한 로봇 플랫폼으로 4족 보행로봇인 제니보를 이용하였으며, 분산 웹 환경에서 동작 가능한 로봇소프트웨어를 개발하기 위해 마이크로소프트사의 MSRS를 이용해 로

봇 기능들을 각각의 서비스로 구현하였다. 실험을 통해 개발된 로봇 서비스들은 분산 웹 환경 하에서 원격 모니터링 및 서비스의 동적인 실행 및 구성이 가능함을 보여 주었으며, 또한 개발된 로봇 서비스를 이용하여 비주얼 프로그램 언어로 로봇어플리케이션을 제작하였다.

특히 본 개발환경에 이용된 MSRS는 다양한 웹 프로그램, 애플리케이션 프로그램이 연동가능하나 마이크로소프트사의 윈도우 계열이 아닌 Linux 등 다양한 운영체제를 지원하는 기능에는 한계가 있다. 하지만 로봇 개발과정 중 시뮬레이션상의 로봇 혹은 기존 로봇으로 표준화된 로봇 서비스들을 이용해 로봇 소프트웨어 개발을 진행할 수 있는 장점이 있다. 또한 분산 웹 환경에서 이종 로봇간의 서비스 상호운용이 가능하다는 점이 특징이다.

본 논문에서 제안된 로봇 서비스 개발환경은 4족형 로봇을 위한 지능로봇기술 개발을 위한 효율적인 연구 및 개발환경을 구축하였다는데 의미가 있다. 나아가 웹 환경에서 로봇의 협업 개발 및 제어가 가능하며, 로봇 서비스를 추상화하여 하드웨어로 인한 제약 사항을 해결함으로써, 다양한 고수준의 보행 로봇에 손쉽게 적용이 가능하다.

참고문헌

[1] ETRI, “로봇 S/W 아키텍처연구팀, “URC 소프트웨어 아키텍처 연구개발 동향”, 2004.

[2] H. Utz, S. Sablatnog, S. Enderle and G. Kraetzschmar, “Miro-Middleware for mobile robot application”, IEEE Transactions on Robotics and Automation, 2002.

[3] B. Gerkey, R.T. Vaughan, and A. Howard, “The Player/Stage project: Tools for multi-robot and distributed sensor systems,” Proceedings of the 11th International Conference on Advanced Robotics, 2003.

[4] Orocos(Open Robot COntron Software) 프로젝트: <http://www.orocos.org>.

[5] 일본 로봇공업회, “RT 오픈 아키텍처와 보급 시스템의 조사연구 성과 보고서,” 2004.

[6] Evolution Robotics, ERSP datasheet, <http://www.evolution.com>.

[7] Wen Services, Oscar Almeida, Johannes Helander,

Henrik Nielsen, Nishith Khantal, “Connecting Sensors and Robots through the Internet by Integration Microsoft Robotics Studio and Embedded”, Proceeding of IADIS International Conference 2007.

[8] Microsoft Robotics Developer Studio, <http://msdn.microsoft.com/en-us/robotics>.

[9] Jiaqi Zhang, Qijun Chen, “Learning based gaits evolution for an AIBO dog”, IEEE Congress on Evolutionary Computation, 2007.

[10] <http://genibo.dasarobot.com/>

[11] Roy Thomas Fielding, “Architectural Styles and the Design of Network-based Software Architectures”, UNIVERSITY OF CALIFORNIA, IRVINE, 2000

[12] <http://cafe.naver.com/msrskorea.cafe>

저자소개

양태규(Tae-kyu Yang)



1982년 광운대학교 전자공학과 학사
1984년 광운대학교 대학원 전자공학과 석사

1989년 광운대학교 대학원 전자공학과 박사
1991년 ~ 2006 목원대학교 전자공학과 교수
2007년 ~ 현재 목원대학교 지능로봇공학과 교수
※관심분야: 로봇제어, 지능로봇

서용호(Yong-Ho Seo)



1999. KAIST 전산학과 학사
2001. KAIST 전자전산학과 석사
2007. KAIST 전자전산학과 박사
2007. Intern Researcher, Robotics Group, Microsoft, US

2008. Consultant, Qualcomm, US
2009. ~ 현재 남서울대학교 컴퓨터학과 전임강사
※관심분야: 지능로봇, 로봇비전, 인공지능