

OSGi 프레임워크 기반의 URC 서버의 구조 제안

김수연, 황석찬, 광동규, 최재영
송실대학교 컴퓨터학부

e-mail : {sykim, schwang, coolman}@ss.ssu.ac.kr, choi@ssu.ac.kr

Proposal for Structure of URC Server Based on OSGi Framework

Sooyeon Kim, Seokchan Hwang, Donggyu Kwak, Jaeyeong Choi
School of Computing, Soong-Sil University

요 약

URC 는 유비쿼터스 환경의 네트워크 기반 로봇이다. URC 로봇은 네트워크를 통해 외부 디바이스에 서비스를 요청하고 기능을 제공 받을 수 있는 환경을 가진다. 이에 따라 URC 로봇을 관리하고 서비스를 제공할 수 있는 외부 디바이스의 중요성이 증가 할 것이다. 본 논문은 유비쿼터스 환경의 URC 로봇 클라이언트의 서비스 요청에 대한 기능을 제공 할 수 있는 외부 디바이스인 URC 서버의 구조에 대해 제안하며, 제안하는 서버는 OSGi 프레임워크를 기반으로 한다.

1. 서론

지능형 서비스 로봇은 다양한 환경에서 네트워크 시스템과 연계하여 인간과 상호작용하며 상황에 따라 주어진 역할을 수행한다[1]. 유비쿼터스 환경에서 동작하는 네트워크 기반의 URC(Ubiquitous Robotic Companion) 로봇[2]은 네트워크를 통해 외부 디바이스와 상호 연동할 수 있다. 이를 통하여 환경 인식과 같은 외부 정보에 대한 처리와 같이 로봇이 수행해야 할 기능을 외부 디바이스에 분담시켜 로봇의 하드웨어 구성을 단순화시켜 원가 절감, 및 외부 디바이스에 서비스 요청을 통한 다양한 서비스를 제공 받을 수 있는 환경을 구축할 수 있게 된다.

이러한 로봇에게 서비스를 제공해 줄 수 있는 외부 디바이스로 URC 서버가 있다. URC 서버는 서버와 클라이언트 로봇이 제공하는 여러 기능을 활용하여 서비스를 구현하고 로봇 클라이언트에서 이를 사용할 수 있도록 서비스 인터페이스를 제공하여야 한다. 서비스 제공을 위한 수행환경은 서비스를 동적으로 제공하는 환경과, 제공된 서비스를 사용하는 로봇 클라이언트에 대한 생명 주기(Life Cycle) 관리와 같은 서비스 제공에 관련된 관리를 위한 환경이 제공되어야 한다. 이를 바탕으로 제안하고자 하는 URC 서버의 구조는 다음과 같다.

- URC 서버는 로봇의 접속 및 이벤트등을 관리 하기 위한 로봇 관리자(Robot Manager)를

가진다.

- URC 서버는 서비스 인터페이스(Service Interface)와 특정 서비스를 제공하는 서비스 제공자(Service Provider)를 가진다.
- URC 서버는 서비스 제공자를 관리하는 서비스 제공자 매니저(Service Provider Manager)를 가진다.

본 논문은 앞서 언급한 환경을 제공하기 위해 OSGi 프레임워크를 기반으로 한다. OSGi[3]는 동적인 서비스 구축 및 관리가 용이한 서비스 플랫폼으로 OSGi 의 구현체인 OSGi 프레임워크에서 제공하는 기능을 이용하여 동적인 로봇 접속 관리 및 서비스 제공을 위한 환경을 구축하는데 이점을 가진다. 이를 통해 OSGi 프레임워크를 기반으로 유동적인 유비쿼터스 환경에서 작동하는 URC 로봇에 대해 로봇이 요구하는 서비스를 제공하기 위한 효율적인 URC 서버의 구조를 제시하고자 하며 논문의 구성은 다음과 같다. 2 장에서는 관련연구에 대해 알아보고, 3 장에서는 제안하고자 하는 URC 서버의 구조 및 기능을 설명하며, 4 장에서는 URC 서버 모델 테스트에 대해 설명한다. 마지막으로 5 장에서는 결론 및 향후 연구 과제에 관하여 논의한다.

2. 관련연구

2.1. URC

URC 는 IT 인프라와 로봇의 융합을 통하여 “언제

"본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음"
(IITA-2008-C1090-0803-0006)

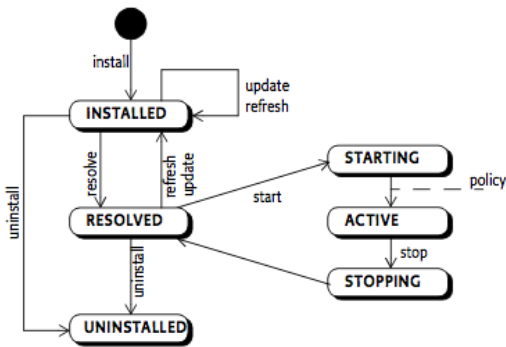
어디서나 나와 함께 하면서 나에게 필요한 서비스를 제공하는 '로봇'을 말한다. 현재 정부에서는 사업단을 구성하여 서비스 사업자, 로봇플랫폼 제조업체, 콘텐츠 및 솔루션 개발업체 등이 유기적 협력 체제를 구축한 상태이다. 로봇 서비스 비즈니스 모델을 확립하고 로봇 사용자의 편익을 극대화하는 등 국민 누구나가 로봇을 손쉽게 이용할 수 있는 토대를 마련하는데 목표를 두고 있다.[2,4]

2.2. RUPI

RUPI 는 로봇 S/W 컴포넌트의 재사용성 및 상호호환성, 다양한 정보기기와의 상호운용성, 이중 통신망과의 상호접속성을 갖는 지능형 로봇의 S/W 규격 및 구현 모델을 말한다.[5] RUPI 규격 중 로봇 내부 태스크 및 콘텐츠 실행 미들웨어의 참조 구현 시스템으로 uROSE[6]가 있다. 본 논문에서 제시하는 URC 서버는 RUPI 규격에서 제시하는 URC 서버의 참조 구현 모델이다.

2.3. OSGi

OSGi[3]는 원격 관리 될 수 있는 자바 기반의 서비스 플랫폼을 제정하는 '개방형 표준 단체'이다. OSGi 는 응용 프로그램의 생명주기(Life cycle) 모델과 서비스 레지스트리(Service Registry)를 정의한다. OSGi 는 OSGi 프레임워크(Framework)를 통해 독립적인 자바 가상 머신 환경에서 제공하고 있지 못한 동적인 컴포넌트 모델을 구현하고 있다. 예를 들어 재시동 과정 없이도 서비스를 모아놓은 개념인 번들을 원격지를 통해 설치(installed), 시작(started), 정지(stopped), 업데이트(updated) 그리고 제거(uninstalled)할 수 있는 일을 수행한다. (그림 1)은 OSGi 번들의 상태 다이어그램을 나타낸다.



(그림 1) OSGi 번들 상태 다이어그램

OSGi 프레임워크를 통해 응용 프로그램 개발에서 가장 복잡하고 관리하기가 어려운, 모듈간의 동적(Dynamic) 관계와 의존(Dependency)을 효과적으로 관리할 수 있게 한다.

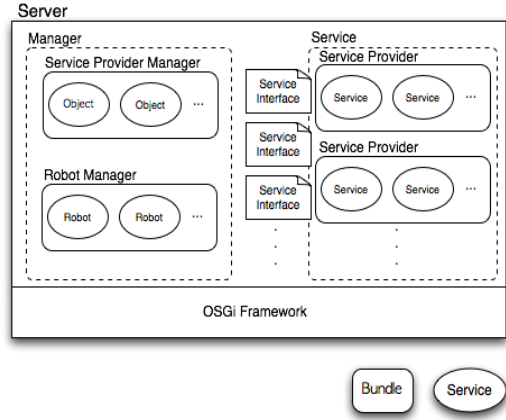
OSGi 아키텍처가 로봇 분야에서 아직 활발하게

사용되고 있진 않으나, 여러 방향에서 연구가 진행 중이다.[7]

3. URC 서버의 구조 및 기능

본 논문에서 제안하는 URC 서버의 구조는 (그림 2)와 같다.

(그림 2)에서 서버는 로봇 관리자(Robot Manager), 서비스 제공자 관리자(Service Provider Manager), 서비스 제공자(Service Provider)와 서비스 인터페이스(Service Interface)로 구성된다.



(그림 2) URC 서버 구조

로봇 관리자(Robot Manager)는 URC 서버에 접속하는 로봇에 대한 정보를 관리한다. 로봇의 접속 상태에 대한 정보와 로봇이 요청하는 서비스를 서비스 제공자 관리자(Service Provider Manager)에 요청하여 해당 서비스 제공자(Service Provider)로부터 서비스 등록 및 등록된 서비스 정보를 제공받아 원격에서 사용할 수 있는 환경을 제공한다.

서비스 제공자 관리자(Service Provider Manager)는 번들로 구성된 서비스 제공자(Service Provider)의 정보를 관리하는 역할을 수행한다. 등록된 서비스 제공자(Service Provider)를 통해 서비스를 등록하고 제공 받을 수 있는 환경을 제공하고, 서비스 제공자(Service Provider)의 등록, 삭제, 시작, 중지 등의 관리를 담당한다. 이를 통해 서버는 동적인 서비스 제공 환경을 구축 할 수 있다.

서비스 제공자(Service Provider)는 특정 서비스 인터페이스(Service Interface)를 구현한 객체를 OSGi 프레임워크에 서비스 형태로 등록한다. 서비스 등록을 통해 서비스 간의 연결 및 제어를 효율적으로 할 수 있는 환경을 제공 할 수 있다.

세가지 구성 요소들은 번들 형태로 OSGi 프레임워크에 등록된다. OSGi 환경에서 번들은 자바 기반의 어플리케이션을 내포하기 위한 개체로서 자바 클래스들과 최종 사용자에게 기능을 제공할 수 있는 다른 리소스들을 포함하고, 다른 번들 들에게 서비스

라고 불리는 컴포넌트를 제공한다.[3]

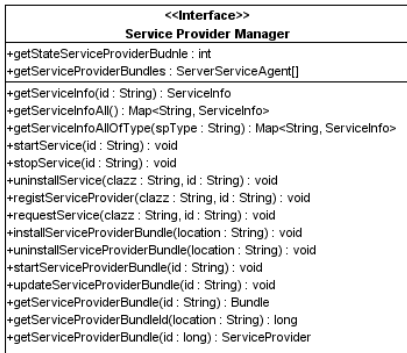
번들로 등록되는 구성요소들은 각각의 기능을 수행하는 객체를 서비스로 등록하여 OSGi 프레임워크(Framework)의 제어 아래서 다른 번들 또는 서비스에서 요청 및 사용 할 수 있는 환경을 구축 할 수 있게 된다. 등록된 서비스는 자신을 등록한 번들의 생명 주기(Life Cycle)에 속하게 된다. OSGi 프레임워크는 서비스들과 서비스 내부의 서비스 객체들을 매핑 하고 서비스들 간의 연결 및 제어를 지원한다. 또한 이벤트 메커니즘을 제공하여 번들의 등록, 수정 및 등록해제 된 서비스 객체의 이벤트를 받을 수 있다. OSGi 환경을 이용하여 URC 서버는 기본적인 기능 수행 외에도 서버 동작 중에 새로운 서비스 제공을 위한 서비스 인터페이스(Service Interface), 서비스 제공자(Service Provider) 등록 또는 서비스 제공자(Service Provider) 수정 시 연관된 디펜던시(Dependency)들과 이벤트 등을 프레임워크를 통해 관리할 수 있게 된다.

4. URC 서버 모델 테스트

본 장에서는 제안한 구조를 기반으로 클라이언트 로봇을 관리 하는 URC 서버 모델을 테스트 한다. 제안하는 서버의 테스트 모델은 x86 기반의 윈도우 XP 에서 개발하였고, Window XP, Mac OS X, Linux 기반의 운영체제에서 테스트 하였다.

OSGi 프레임워크(Framework)는 OSGi Specification Release 4.0 을 지원하는 eclipse Equinox[8]를 이용하였다.

(그림 3)은 서비스 제공자 매니저(Service Provider Manager)를 위한 클래스 다이어그램이다.

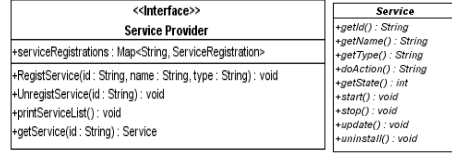


(그림 3) Service Provider Manager 클래스 다이어그램

서비스 제공자 매니저(Service Provider Manager)는 번들 형태로 등록되는 서비스 제공자(Service Provider) 관리를 위한 부분을 정의한다. 서비스 제공자 매니저(Service Provider Manager)는 서비스 제공자(Service Provider)를 OSGi 프레임워크에

등록하여 그 정보를 관리하고, OSGi 프레임워크의 관리하에 서비스 제공 환경을 효율적으로 구축한다.

(그림 4)는 서비스 제공자(Service Provider)와 서비스를 위한 클래스 다이어그램이다.



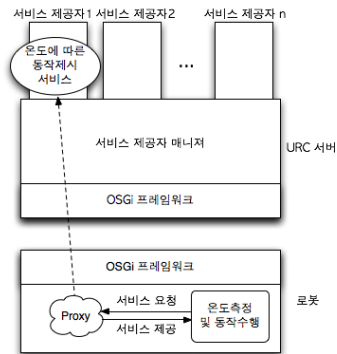
(그림 4) Service Provider 와 Service 클래스 다이어그램

서비스 제공자(Service Provider)는 특정 서비스 인터페이스(Service Interface)를 구현한 서비스를 OSGi 프레임워크에 등록, 해제 등의 서비스 사용에 관련된 기능을 제공 한다. 서비스를 등록하고 그 정보를 관리하여 서비스 제공과 제어를 효율적으로 할 수 있는 환경을 제공한다.

본 논문에서 제안하는 URC 서버 모델의 테스트 시나리오는 다음과 같다.

- 1) 원격에서 로봇이 접속하여 특정 서비스를 서버에 요청한 상황이라 가정한다.
- 2) 온도를 측정하여 특정 온도에 대해 창문을 여닫는 동작을 수행하는 로봇이 온도를 측정하여 전송하여 서버 측에 동작을 요청하는 서비스를 요청하고자 한다.
- 3) 서버는 클라이언트 로봇의 정보를 받아 특정 동작을 제공해주는 서비스를 로봇을 대신하여 수행한 뒤 로봇에게 제공한다.
- 4) 클라이언트 로봇은 클라이언트 내부에서 서비스를 수행 하는 것처럼 서버로부터 수행된 서비스를 통해 동작을 지시 받아 창문을 여닫는다.

(그림 5)는 URC 서버 모델의 테스트 구조를 나타낸다.



(그림 5) URC 서버 모델의 테스트 구조

(그림 5)의 구조를 바탕으로 로봇에서의 온도 측정 정보가 40° C 이상일 때 서버에서 동작을 제시할 때 창문을 여는 명령을 내리고, 이외의 경우에는 창문을 닫는 명령을 내리도록 설정한다. 이를 통해 URC 서버는 클라이언트 로봇을 대신해서 동작을 제어하는 서비스를 클라이언트에게 제공해 줄 수 있다.

<표 1> URC 서버 모델 테스트 실험 결과 -서버-

이벤트	문자열 출력
1.b 서비스 요청 처리	requestService - window : RobotA
1.c 서비스 등록	registService - window : RobotA
2.b 온도측정정보수신	receiveInfo - 45 : RobotA
3.a 명령 전달	doAction

<표 2> URC 서버 모델 테스트 실험 결과 -가상 클라이언트 로봇-

이벤트	문자열 출력
1.a 서비스 요청	requestService - window : RobotA
2.a 온도측정정보송신	sendInfo - 45 : RobotA
3.b 명령 실행	doAction - open the window

실험 결과는 URC 서버 및 가상 클라이언트 로봇 모두 Equinox 프레임워크에 번들 형태의 모듈을 설치하여 이벤트 발생 시 해당 이벤트 처리 시 문자열 출력을 하는 환경으로 구성하였다. <표 1>과 <표 2>는 이벤트에 따른 문자열 출력 결과를 통해 본 논문에서 제안한 구조를 실험한 결과를 보여준다. 가상 클라이언트 로봇이 특정 서비스를 서버에 요청 하고, 서버가 동적으로 서비스 등록 후 로봇을 대신하여 서비스를 수행하며 특정 명령을 네트워크를 통해 로봇에게 전달하여 로봇이 명령을 실행하는 과정을 잘 수행하고 있다. 서버는 Mac OS X 환경에서 실행하고, 가상 클라이언트 로봇은 Windows XP 환경에서 실험을 진행 하였다. URC 로봇의 Proxy 서비스는 R-OSGi[9]를 이용하여 실험하였다.

5. 결론 및 향후 연구과제

본 논문에서는 OSGi 프레임워크 기반의 URC 서버의 구조에 대해 제안하였다. URC 로봇의 서비스 요청에 대한 처리를 하는 URC 서버의 구조 및 기능은 다음과 같다.

- 로봇 관리자(Robot Manager)
 - 로봇의 접속관리 및 이벤트 처리
- 서비스 제공자 매니저(Service Provider Manager)
 - 동적인 서비스 제공자 관리
 - 서비스 제공 환경 제공
- 서비스 인터페이스(Service Interface)
 - 특정 서비스에 대한 정의
- 서비스 제공자(Service Provider)
 - 특정 서비스 생성 및 제공을 위한 관리

본 논문에서 제시한 URC 서버의 구조는 OSGi 프레임워크를 기반으로 하였다. OSGi 프레임워크는

사용자의 다양한 요구를 만족시키기 위해서 서비스 추가, 삭제 등의 동적 서비스 관리가 가능하다. 이를 통해 URC 서버 구축 시 유비쿼터스 환경의 URC 로봇이 동적으로 URC 서버에 접속 및 서비스 요청 시 서비스 제공 환경을 구축하고 관리하는데 큰 이점을 가진다. 현재 제안한 구조에서 각 모듈간의 효율적인 이벤트 관리, 모니터링 기능 등에 대해 연구가 진행 중이며, 앞서 언급한 추가적인 기능에 대해 동적으로 설치 및 환경구축을 할 수 있는 서버 구조를 통해 효율적인 서버 관리를 위한 기능을 제공할 예정이다.

참고문헌

- [1] 김성훈, 김중배, “URC 를 위한 로봇 S/W 아키텍처 기술”, 대한전자공학회 특집호, 제 33 권, 제 3 호, pp.56-63, 2006.
- [2] URC, <http://www.urckorea.com>
- [3] OSGi, <http://www.osgi.org/>
- [4] 조영조, 오상록, “지능형 서비스 로봇과 URC”, 한국통신학회지(정보와 통신), 제 21 권, 제 10 호, pp.13-21, 2004.
- [5] RUPI, <http://www.rupi.or.kr/>
- [6] 정승욱, 이승익, 김성훈, “네트워크 로봇을 위한 로봇 소프트웨어 플랫폼에 대한 연구”, 정보과학회지, 제 26 권, 제 4 호, pp.38-48, 2008.
- [7] Shengyuan Luo, Ping Jiang, Jin Zhu, “Software Evolution of Robot Control Based on OSGi” Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on Publication Date: 22-26 Aug. 2004 On page(s): 221-226
- [8] Equinox OSGi, <http://www.eclipse.org/equinox/>
- [9] R-OSGi, <http://r-osgi.sourceforge.net/>