URC 로봇 상태정보 모니터링을 위한 SOMAR 환경에서의 분산 Observer 패턴 구현

김수연, 곽동규, 최재영, 유재우
숭실대학교 컴퓨터학과
e-mail : sykim@ss.ssu.ac.kr

Implementation of Distributed Observer Pattern about SOMAR environment for URC Robot Status Information Monitoring

Sooyeon Kim, Donggyu Kwak, Jaeyoung Choi, Chae-Woo Yoo
School of Computing, Soongsil University

요 약

URC 로봇은 고성능 서버를 통해 로봇이 수행할 서비스의 일부를 외부 디바이스에 분담시킴으로써 저렴한 하드웨어로 로봇을 구현하고자 하는 목적을 가지는 지능형 서비스 로봇이다. SOMAR는 유비쿼터스 환경에서 사용자에게 다양한 원격 로봇 서비스를 제공하기 위한 URC 로봇 미들웨어로, URC 로봇 클라이언트와 이를 등록, 제어, 관리하는 서버로 구성된다. SOMAR 서버를 통해 로봇 서비스를 수행하기 위해서는 로봇 클라이언트의 상태정보를 모니터링하여 수행가능 여부를 판단해야 한다. 본 논문은 이러한 상태정보 모니터링을 위해 R-OSGi를 이용한 분산 Observer 패턴을 구현한다. Observer 패턴은 객체간에 느슨한 결합구조를 가져 URC 로봇 클라이언트 모니터링 프로그램의 개발과 수정이 용이하고, 플러그인 방식의 개발을 통해 다양한 프로그램 설치와 응용이 가능하다.

1. 서론

URC (Ubiquitous Robotic Companion)는 "언제 어디서나, 나와 함께 하며, 나에게 필요한 서비스를 제공하는 로봇"으로, 고성능의 서버를 통해 로봇이 필요하거나 수행하기 어려운 다양한 로봇 서비스를 수행하여 로봇 하드웨어 구성을 단순화시킨다. 이를 통해 로봇 개발 단가를 절감하고 다양한 서비스를 제공할 수 있다[1][2]. SOMAR (Service Oriented Middleware Architecture for Robot)는 URC 로봇 클라이언트가 서버를 통해 다양한 로봇 서비스를 수행할 수 있는 서비스 지향적 미들웨어 아키텍처이다[3]. URC 로봇 클라이언트는 서버에 접속해 로봇 서비스 수행을 위한 관리를 제공 받게 된다. 따라서 서버는 로봇 클라이언트의 상태를 모니터링하며 로봇 서비스 수행 여부를 판단할 수 있어야 한다.

본 논문은 SOMAR 환경에서 로봇의 상태정보를 모니터링하고 제어하기 위한 URC 로봇 클라이언트 상태정보 모니터링에 대해 다룬다. 상태 정보 모니터링을 위해 Observer 패턴을 응용하며 R-OSGi [4][5]를 이용해 분산 환경에서의 이벤트 통지를 수행한다. 본 논문에서 제안하는 방식은 객체간의 느슨한 결합구조를 통해 세부사항을 교려할 필요가 없게 되므로 응용 프로그램의 개발 및 수정이 용이하고, 재사용성을 증가시킨다. 또한 응용 프로그램의 다양성과 확장성을 향상시킬 수 있다. 본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 설명하고, 3장에서 로봇 클라이언트 상태정보 모니터링에 대해 논하고, 4장에서 본 논문의 결론을 정리한다.

2. 관련연구

2.1 로봇 모니터링

로봇의 원격 모니터링 기술은 원격에서 제어 및 모니터링을 지원하는 기술로, 로봇 내부 이벤트 정보의 모니터링 및 원격 제어가 가능하도록 하는 서버 기반의 제어 기술이 요구된다[6]. 하지만, 기존의 모니터링 방식은 각 로봇 특성에 한정적이어서 다양한 로봇에 동일한 방법으로 적용하기 어렵다. 본 연구는 이기종의 로봇에 대해 모니터링 시스템의 적용이 용이하다.

2.2 모바일 M/VC 응용 프레임워크

모바일 M/VC 응용 프레임워크는 자바의 Observer를 무선 환경으로 확장한 MobileObserver 클래스를 정의하여 논리적 레벨에서 상호작용을 한다. 물리적 레벨에서의 통신 및 상호작용은 Observable 측의 Multiplexer와 Observer 측의 Demultiplexer에 의해 이루어진다. Observable는 Multiplexer를 통해 이벤트를 보내고 Observer는 Demultiplexer를 통해 변경된 값을 전달받는다. 이런 모바일 M/VC 응용의 객체들 간의 메시지 흐름은 Multiplexer와 DeMultiplexer에 집중 된다[7].

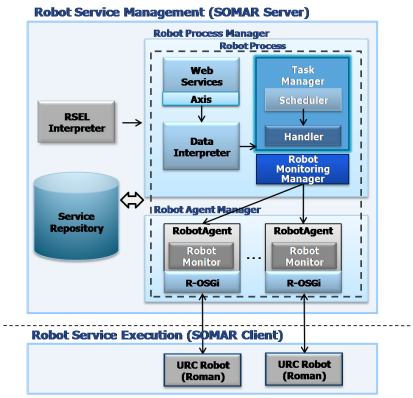
3. 로봇 클라이언트 상태정보 모니터링

3.1. SOMAR 시스템 구조

그림 1은 SOMAR의 시스템 구조를 보여준다. SOMAR 시스템 구조는 URC 로봇 클라이언트와 이를 연결, 관리하는 서버로 구성된다. 각 계층은 로봇 클라이언

"본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음"
(IITA-2009-(C1090-0902-0007))

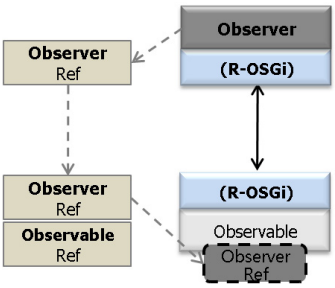
트가 서버를 통해 로봇 서비스를 수행하기 위해 필요한 연결 구조를 보여준다. SOMAR 계층 구조를 통해 서버는 접속된 로봇 클라이언트의 상태 정보를 모니터링하여 로봇 서비스 수행 여부를 판단할 수 있다.



(그림 1) SOMAR 시스템 구조

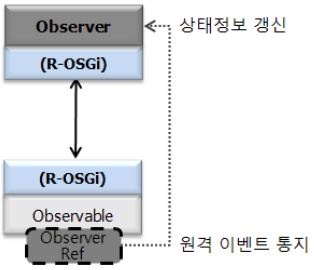
3.2. 분산 환경에서의 Observer 패턴 구현

그림 2는 Observer 패턴을 응용하여 URC 로봇 클라이언트의 상태정보를 모니터링하기 위한 구조를 보여준다.



(그림 2) 로봇 클라이언트 상태정보 모니터링 구조

서버는 Observer 객체를 관리하며, 로봇 클라이언트 접속시 Observer 객체 reference를 전달한다. 로봇 클라이언트에서 전달받은 객체 reference는 Observable 객체에 등록한다. 이벤트 발생 시 Observable 객체는 등록된 Observer 객체 reference에 이를 통보한다. 이때 원격 서비스인 Observer 객체에 접근하기 위해서 R-OSGi를 이용한다. R-OSGi 서비스로 등록된 Observer 객체 reference를 통해 Observable 객체에서 proxy 형태의 원격 서비스를 호출할 수 있다.



(그림 3) R-OSGi를 이용한 원격 이벤트 통지

그림 3은 R-OSGi를 이용해 분산 환경에서 Observer 패턴을 응용하는 구조를 보여준다. R-OSGi의 원격 서비스 호출을 통해 Observable 객체에 등록된 Proxy Observer 객체는 Observable 객체에서 이벤트 통지 시 원격 서비스 호출을 통해 상태정보를 갱신하여 모니터링을 수행한다. 로컬에서 직접적으로 원격 서비스를 호출할 수 있게 되므로, 추가적인 이벤트 전달 처리를 할 필요가 없다. 각 객체는 상호작용을 추상화한 인터페이스를 구현하여 플러그인 방식으로 설치 및 관리가 된다. 이를 통해 재상용성을 높이고 개발을 쉽고 효과적으로 할 수 있다.

4. 결론

본 논문에서는 SOMAR 환경에서 로봇 클라이언트의 상태 정보를 모니터링 하기 위해 R-OSGi를 이용한 분산 환경에서의 Observer 패턴 구현에 대해 다루었다. URC 로봇 클라이언트가 다양한 로봇 서비스를 수행하기 위해서 서버는 로봇 클라이언트의 상태를 모니터링하며 로봇 서비스 수행이 가능한지를 확인해야 한다. 이를 위해 R-OSGi를 이용한 Proxy 형태의 원격 서비스 호출을 통한 분산 환경의 Observer 패턴을 구현하여 상태정보를 모니터링한다. R-OSGi의 proxy 형태의 원격 서비스 호출을 통해 Observable에서 원격으로 Observer에 이벤트를 통보하고 상태정보를 갱신할 수 있다. 인터페이스를 구현한 Observer 와 Observable 객체는 플러그인 방식으로 개발하여 설치 및 삭제가 편리한 장점을 가지고, 각 객체의 개발과 수정이 용이해 확장을 통한 다양한 형태의 모니터링을 할 수 있게 된다.

참고문헌

[1] URC, http://www.urckorea.com
[2] 김성훈, 김중배, “URC를 위한 로봇 S/W 아키텍처 기술”, 전자공학회지 제33권 제3호, pp. 56-63, 2006. 3.
[3] 김수연, 곽동규, 최재영, “URC 로봇에 서비스 제공을 위한 SOMAR 미들웨어의 구조 설계”, 2009 한국컴퓨터종합학술대회 논문집 제36권 제1호, pp.378-382, 2009. 6.
[4] R-OSGi, http://r-osgi.sourceforge.net
[5] J. S. Rellermeyer, G. Alonso, and T. Roscoe. “R-OSGi: Distributed Applications Through Software Modularization”, In Proceedings of the ACM/IFIP/USENIX 8th International Middleware Conference, 2007.
[6] 김주만, “URC 로봇 원격 모니터링기술 개발”, 한국콘텐츠학회논문지 제6권 제8호, pp. 8-19, 2006.
[7] 음두현, “관찰자/피관찰자 설계 패턴을 이용한 모바일 M/VC 응용 프레임워크”, 인터넷정보학회논문지 제7권 제2호, pp. 81-92, 2006.
[8] Karl Ridgeway, David Bernstein, John Magnotti, “Design and Implementation Considerations in a Distributed Observer System”.