

JavaRmi 분산객체를 위한 이벤트서비스의 설계

김 현 규* • 유 양 우 • 박 양 수 • 이 명 준
울산대학교 전자계산학과

A Design of Event Service for JavaRmi Distributed Objects

Hyun-Gyu Kim* • Yang-Woo Eu • Yang-su Park • Myung-Joon Lee
Dept. of Computer Science, University of Ulsan

대규모의 분산응용시스템은 일반적으로 다수의 구성요소로 이루어져 있으며, 그 구성요소간의 상호작용을 위해 필요한 이벤트의 전달 구조가 다소 복잡한 특성을 지닌다. 이러한 이벤트의 다중처리 문제는 난이도가 높은 구현기법 중 하나이므로, 보다 안정적인 분산응용시스템을 효율적으로 개발하기 위해서는 이벤트의 다중처리를 위한 기반 시스템을 제작, 지원하는 것이 바람직하다.

본 논문에서는 Java 에서 분산객체개념을 도입한 JavaRmi 환경에서 이벤트의 체계적인 다중처리를 위한 서비스를 설계함으로써, Java 개발자가 보다 안정적이고 효율적인 방법으로 분산응용시스템을 구축할 수 있도록 지원하기 위한 방법을 제안하고자 하였다. 본 논문에서 제안한 JavaRmi 이벤트서비스 시스템은 JavaRmi 를 기반으로 하여, JavaRmi 시스템이 지닌 효율성과 대중성의 잇점을 충분히 활용할 수 있도록 설계되었다. 이러한 기본적인 잇점 이외에도 이벤트서비스의 설계에 있어서 직관적인 인터페이스, 그룹별 이벤트서비스의 생성과 관리 기능, 병행성과 동기화 도구 등의 기능을 추가하여, 분산응용시스템의 개발시 자주 요구되는 이벤트 처리를 위한 Java 개발자의 노력을 최소화할 수 있도록 지원하고자 하였다.

1. 서 론

90 년대에 이르러 대두된 분산환경은 네트워크로 연결된 기기종 시스템들간의 자원공유를 가능하게 하여 정보이용의 효율을 극대화시키며, 다양한 분산환경 서비스의 투명성을 제공하고, 일부 시스템의 결함에 대처하여 높은 신뢰도를 제공할 수 있는 장점을 지니고 있다[7]. 그러나 분산환경을 구성하는 각 시스템간의 이질성과 원격거리 통신에 대한 세부적인 구현사항은 분산응용시스템을 구축하는 개발자들에게 많은 어려움을 주게 되었다. 이를 극복하기 위하여 소개된 분산객체 컴퓨팅(Distributed Object-Oriented Computing) 기법은 클라이언트/서버 기법과 객체지향 개념을 도입하여 분산환경이 지닌 세부구현 사항으로부터 독립적으로 시스템을 설계하기 위한 방법론으로서, 이 개념을 수용한 시스템의 대표적인 예로는 Sun 사의 JavaRmi[4]와 OMG 의 CORBA[5]를 들 수 있다.

Sun 사의 JavaRmi 는 Java[1]를 기반으로 하여 대중성과 효율성의 측면에서 CORBA 에 비해 우위를 가질 수 있으며, 기존의 다른 언어로 작성된 코드를 재사용하기 위한 JNI 기능이나 데이터베이스와 연결하기 위한 JDBC 등의 기능을 기본적으로 내장함으로써 효율성이 커지고 있다. 그러나 JavaRmi 에서는 JavaRmi 분산객체간에 이벤트의 복잡한 상호작용을 처리하기 위한 기능을 제공하지 않으며, JavaRmi 를

이용하여 대규모의 분산응용시스템을 보다 정교하고 안정성있게 작성하기 위해서는 이러한 이벤트의 다중처리를 위한 기반 시스템이 제공되어야 할 것으로 예측된다.

이에 반해 분산객체컴퓨팅의 표준으로 OMG 에 의해 제안된 CORBA 에서는 이러한 메시지의 다중처리 문제를 체계적으로 지원하기 위해 이벤트서비스(Event Service)[3,6,8]를 정의하고 있다. 그러나 CORBA 이벤트서비스는 그룹 프로그래밍에 있어서 가장 기본적인 이벤트의 다중처리 기능에 대한 추상적인 명세만을 제공하고 있으므로, 이벤트서비스를 사용하더라도 실제 분산응용시스템의 구현시에는 개발자의 별도의 노력을 필요로 하게 된다[2].

따라서 본 논문에서는 CORBA 이벤트서비스의 개념을 보완한 JavaRmi 분산객체를 위한 이벤트서비스의 설계에 대해 소개하며, 이로서 분산응용프로그램의 작성시에 그룹 프로그래밍을 위한 Java 개발자의 노력을 줄일 수 있도록 지원하고자 한다. 이벤트서비스의 기능 지원에 있어서 JavaRmi 를 하부구조(infra-structure)로 사용함으로써 가질 수 있는 잇점은 CORBA 이벤트서비스에 비해 이벤트의 처리 과정이 효율적이며, 분산환경의 세부구현 사항에 독립적으로 구현할 수 있고, 또한 Java 가 지니는 범용성으로 인해 대중화를 보다 쉽게 기대할 수 있다는 것이다. 이러한 기본적인 잇점 이외에도 위에서 제시한 CORBA Event

Service 의 기능을 보완하여 직관적인 인터페이스, 그룹별 이벤트서비스의 생성과 관리 기능, 병행성과 동기화 도구등에 관한 기능을 추가하여 Event Service 의 유연성 (flexibility)을 증가시킴으로써 그룹 프로그래밍을 위한 개발자의 별도의 노력을 줄일 수 있도록 설계한다.

2. 분산객체를 위한 이벤트서비스의 바람직한 기능

이 장에서는 CORBA 이벤트서비스의 개념에 대한 소개와 이를 토대로 CORBA 이벤트서비스에서 기능을 확장한 JavaRmi 분산객체를 위한 이벤트서비스의 기능설계에 대해 언급한다.

2.1. CORBA 이벤트서비스

CORBA 이벤트서비스는 CORBA 객체 간의 이벤트의 다중전이 문제를 체계적으로 지원하기 위한 명세로서, CORBA 객체간의 서비스를 확장하기 위한 명세서인 COSS(CORBA Object Service Specification)[3] 내에서 정의하고 있다. CORBA 이벤트서비스는 이벤트를 생성하는 공급자(Supplier)와 이벤트를 처리하는 소비자(Consumer), 그리고 다수의 공급자와 다수의 소비자간의 중재자 역할을 수행해 주는 이벤트 채널(Event Channel)로 구성된다. CORBA 이벤트서비스에서 이벤트 채널을 통해 이벤트를 전달하고 가져올 때에는 공급자나 소비자가 서로에 대한 정보를 알 필요가 없으며, 이를 통해 CORBA 이벤트서비스는 공급자와 소비자 간에 다중적이고 결합도가 낮은 통신 원리를 제공하게 된다.

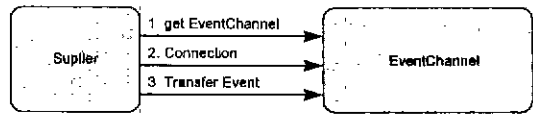
그러나 CORBA 이벤트서비스는 이질적이고 다양한 분산 객체에 광범위하게 적용될 수 있도록 명세를 지나치게 추상화함으로써, 그룹 프로그래밍에 이벤트서비스를 사용하더라도 별도의 이벤트 처리기법을 필요로 하게 되며, 사용에도 어려운 단점이 있다. 따라서 이벤트서비스 시스템이 보다 다양한 영역에서 사용되기 위해서는 이러한 단점에 대한 기능이 보완되어야 할 것으로 예측된다.

2.2. JavaRmi 이벤트서비스의 기능 설계

다음에서는 CORBA 이벤트서비스 개념을 기초로 JavaRmi 이벤트서비스 시스템에서 지원되어야 할 기능에 대한 설계내용을 다룬다.

가. 사용 절차

JavaRmi 이벤트서비스에서는 CORBA 이벤트서비스의 대리자(proxy)에 연결하기 위한 복잡한 상호작용에 대한 구현 부분을 이벤트서비스 내부적으로 처리함으로써, 사용자 인터페이스를 보다 직관적으로 구성하였다. 다음은 JavaRmi 이벤트서비스에서 공급자로서 Event-Channel 에 연결하기 위해 필요한 절차를 소개하고 있다[그림.1].

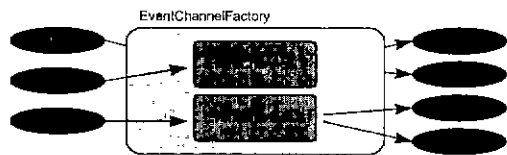


[그림.1] JavaRmi 이벤트서비스를 사용하기 위한 절차

JavaRmi 이벤트서비스에서는 단지 EventChannel 객체의 레퍼런스를 얻어와서 멤버로 연결한 후 send 메소드를 통해 이벤트를 전달할 수 있도록 구성함으로써, 보다 쉬운 인터페이스를 제공하고자 하였다.

나. 그룹별 이벤트서비스의 생성과 관리

CORBA 이벤트서비스에서는 그룹별로 이벤트서비스를 생성하여 사용하기 위한 명세를 정의하지 않으므로 이벤트서비스를 사용하더라도 그룹 프로그래밍을 위해서는 개발자가 별도로 구현을 해야 했다. JavaRmi 이벤트서비스는 이러한 문제를 위해 이벤트서비스내에 기본적으로 Factory 개념을 도입함으로써 그룹 단위의 이벤트서비스 생성이 가능하도록 한다.[그림.2]



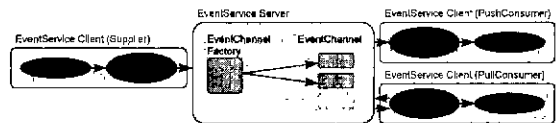
[그림.2] EventChannelFactory

다. 이벤트의 병행 처리와 동기화 기능

JavaRmi 이벤트서비스 시스템은 내부적으로 IDL/SSO 시스템[7]을 이용하여 구현함으로써 이벤트의 병행 처리나 동기화 기능의 구현에 대한 신뢰성을 체계적으로 지원하며, IDL/SSO 시스템이 제공하는 세마포어나 조건변수 등의 고수준 동기화 도구를 이용하여 사용자의 목적에 맞는 동기화 역시 구현할 수 있다.

3. JavaRmi 이벤트서비스의 설계

JavaRmi 이벤트서비스는 그룹별로 이벤트서비스를 제공하는 이벤트서비스 서버와 이벤트를 공급하기 위한 공급자, 공급자로부터 전달된 이벤트를 사용하는 소비자로 구성되어 있으며, 각 공급자와 소비자는 이벤트서비스와 보다 쉬운 인터페이스를 통해 연결하기 위해 대리자(Proxy)를 이용할 수 있도록 설계되었다.

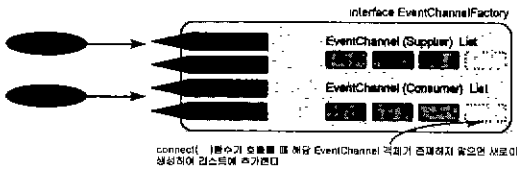


[그림.3] JavaRmi 이벤트서비스의 구조도

JavaRmi 이벤트서비스 서버는 그룹 단위의 이벤트 서비스를 생성하기 위한 EventChannelFactory 객체와

이벤트서비스 고유의 기능인 소비자와 공급자 간의 이벤트의 다중전달기능을 제공하는 EventChannel의 두 가지 객체로 구성된다.

EventChannelFactory 객체의 인터페이스에 대한 기본설계는 [그림.4]와 같다. EventChannelFactory 객체는 그룹 단위로 이벤트서비스를 생성하고 관리하기 위한 기능을 제공하는 객체로서, 내부적으로 공급자와 소비자별로 EventChannel에 대한 리스트를 가지고 관리하게 되며, 새로운 EventChannel을 생성하거나 삭제하는 기능, 현재 등록된 EventChannel을 참조할 수 있는 기능, 특정 EventChannel의 소비자에게 이벤트를 보내기 위한 기능 등을 제공하게 된다.



[그림.4] "EventChannelFactory" 인터페이스

EventChannel 객체는 이벤트서비스 고유의 기능인 소비자와 공급자 간의 이벤트의 다중전달 기능을 제공하는 객체로서 인터페이스에 대한 기본적인 설계는 [그림.5]와 같으며, EventChannel 객체에서는 사용자가 EventChannel에 접속하거나 해제하고, 또한 자신과 연결된 소비자에게 이벤트를 전달하기 위한 인터페이스를 정의하게 된다.



[그림.5] "EventChannel" 인터페이스

EventChannel 객체는 connect() 오퍼레이션을 통해 연결된 멤버에 대한 리스트를 유지하며, send()를 통해 전달된 메시지는 연결된 모든 멤버에게 전달된다. 그리고 각 멤버에서는 자신에게 전달된 메시지를 다시 소비자에게 보냄으로써, 이벤트의 다중전달 기능을 구현하게 된다.

4. 결 론

본 논문에서는 CORBA 이벤트서비스 개념을 발전시켜 JavaRmi 분산객체를 위한 이벤트서비스의 개발에 도입함으로써, Java 분산객체 응용 프로그램의 작성시에 이벤트의 다중 처리를 위한 Java 개발자의 노력을 최소화하는 이벤트 서비스의 설계를 제안하였다. 설계된 JavaRmi 이벤트서비스는 JavaRmi를 하부 구조(infrastructure)로 사용함으로써 CORBA 이벤트서비스에 비해 이벤트의 처리 과정에 있어서 효율적이며,

분산 환경의 세부 구현 사항에 독립적으로 구현할 수 있고, 또한 Java가 지니는 범용성으로 인해 쉬운 대중화를 기대할 수 있다. JavaRmi 이벤트서비스에서는 쉬운 인터페이스와 Factory 개념을 통한 그룹별 이벤트 서비스의 생성과 관리기능, 이벤트의 병행처리(concurrency) 기능과 동기화(synchronization) 도구를 지원하여 그룹 프로그래밍을 위한 개발자의 별도의 노력을 줄일 수 있도록 하였다.

설계된 JavaRmi 이벤트서비스는 현재 구현중에 있으며, 정보공유를 목적으로 한 그룹 분야의 대부분의 분산 응용 시스템 작성에 적용되어 Java를 기반으로 한 사용자들의 호응을 얻을 수 있을 것으로 기대된다. JavaRmi 이벤트서비스 시스템은 시스템 관리 도구, 특성 리스트 객체, CASE(Computer Aided Software Engineering) 도구, 스프레드 시트, 주식 시세 통보 시스템 등의 주로 이벤트의 공급자와 소비자 간의 느슨한 연결 관계를 지닌 그룹 소프트웨어 분야에 광범위하게 적용될 수 있다.

현재 Java 언어와 같은 순수 범용 언어(host language) 수준에서 직접적으로 다중 이벤트 처리를 지원하는 연구나 상품은 전무한 상황이므로, JavaRmi 이벤트서비스에 대한 연구는 추후의 다른 JavaRmi 분산객체를 위한 이벤트서비스에 대한 연구에도 도움을 줄 수 있을 것으로 예측된다.

5. 참고문헌

- [1] "The Java Language Tutorial", Sun Microsystems, Inc., 1995
- [2] Douglas C. Schmidt, Steve Vinoski, "Distributed Callbacks and Decoupled Communication in CORBA (Column 8-10)", SIGS C++ Report Magazine, February 1997
- [3] Object Management Group, "CORBAServices-Common Object Services Specification", OMG 95-3-31, 1995
- [4] "JavaRmi: Java Remote Method Invocation", Sun Microsystems, Inc, 1996
- [5] Object Management Group, "Common Object Request Broker: Architecture and Specification", 1995
- [6] IONA "OrbixTalk White Paper: The CORBAService Event Service", IONA Technologies Ltd., 1996
- [7] 박양수, 김현규, 이명준, 한상영, "A Specificaiton Language System for Shared Object under CORBA Open Distributed Computing Environment", 한국정보처리학회 논문지 제 5 권 2 호 pp.404-414, May 1998.
- [8] 정혜영, 김현규, 박양수, 구자복, 이명준, "자바를 이용한 CORBA 이벤트서비스의 개발", 한국정보과학회 가을 학술발표논문집 Vol.24, No.2 pp.237-240, October 1997.