

이동객체 ORB 시스템의 개발

조애향^o, 유기영

경북대학교 컴퓨터공학과

ORB System Supporting Mobile Object

AeHyang Cho and KeeYoung Yoo

Department of Computer Engineering Kyungpook Nat'l Univ.

요 약

이동 에이전트를 위한 이동객체(mobile agent)와 이를 지원해 주는 이동객체 서버(mobile object server)를 설계하고 이러한 요소들을 제어하고 관리하는 이동객체 시스템(mobile object system)을 Java로 구현한다. 또, 시스템내의 이동객체를 비롯한 모든 객체들간의 객체 요청(object request)을 유연하게 증계하고 이들간의 통신을 담당하는 ORB(Object Request Broker)를 TCP/IP상에서 Java로 구현하고 이동 에이전트 시스템을 쉽게 구축할 수 있는 앱스트랙션(abstraction)과 API를 제공한다 이 시스템은 네이밍 서비스(naming service)와 푸쉬형 이벤트 서비스(push-event service)등을 기본적으로 지원한다.

1. 서론

네트워크 기술의 발전으로 네트워크 중심의 프로그램에 대한 관심이 많아지고, 이에 호응하여 데이터 이동만으로 서비스를 제공하던 기존 통신 응용 프로그램들은 서버의 과부하를 줄이고, 네트워크 이벤트의 발생을 줄이면서 필요한 서비스를 사용자에게 지원하기 위한 하나의 방법으로 데이터와 코드를 결합한 객체의 이동 즉, 이동 에이전트를 통하여 서비스를 지원하는데 대한 관심이 높아지고 있다. 최근의 객체의 재사용성에 중점을 둔 객체 지향 기술의 중요성과 분산처리의 경향에 더불어 이동성을 가진 객체에 대한 관심도 더불어 대두되고 있다. 즉, 이동 에이전트(mobile agent)를 객체 지향적인 관점에서 지원하는 이동객체와 이동객체 시스템에 대한 요구가 대두되고 있다.

이동객체를 기존의 프로그래밍언어로 구현하기에는 기존 언어의 프로그램은 극도로 플랫폼에 의존적이고, 번역된 프로그램 코드들은 한 호스트 내의 주소 영역(address space)에서 실행되므로 다른 호스트간의 코드이동 및 실행을 위해서는 프로그래머가 이를 별도의 프로그램으로 관리해야 하는 등의 어려움이 있다. 이동객체를 구현하기가 어렵고, 비용이 많이 들며, 투명성이 제한되어 객체 모델에 한계가 있고, 객체의 이동성을 지원하는 하부구조(infrastructure)의 부족[1] 등의 이유로 이동객체에 관한 관심에도 불구하고 실제 크게 활용화 되지 못하였다

이러한 이동객체 시스템은 Java 개발로 크게 발전하게 되었다.

즉, Java의 기계독립성(machine-independent)을 가진 JVM, 클래스 로딩(class loading) 모델 그리고 기타 RMI(Remote Invocation Interface)에 관련된 기능으로 이동 에이전트 시스템(mobile agent system) 분야에 많은 연구와 발전이 있었다[2].

본 연구에서는 Java 기술을 이용하여 이동객체를 지원하고, 기존의 객체 모델을 수용하고, 이기종의 시스템에도 잘 적용이 되는 이동객체 시스템을 개발하였다. 이 시스템은 JVM 상에서 객체들 간의 통신을 유연하게 지원하는 ORB(Object Request Broker)를 지원하고 ORB로서 요구되는 기본적인 서비스를 지원한다. 이 글은 2장에서는 이동 에이전트 시스템의 구성과 시스템 요구사항에 대해 설명하고, 3장에서는 이러한 요구사항을 따라 구현한 이동객체 시스템, 각 구성 요소와 서비스에 대해 설명하고 4장에서는 결론은 맺는다.

2 이동 에이전트 시스템

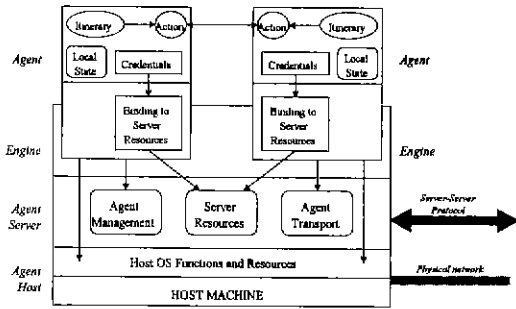
2.1 개요

넓은 의미에서 에이전트는 사용자 즉 사람을 대신해서 작업을 하는 모든 프로그램을 일컬으며 특히, 이동 에이전트는 네트워크 상에서 사용자를 대신해서 지정된 일정에 따라 노드와 노드 사이를 자치적으로(autonomously) 이동할 수 있는 프로그램을 말한다. 응용 프로그램은 이동 에이전트를 네트워크에 투입하고, 에이전트는 네트워크를 미리 정해진 일정(itinerary)에 맞추어 움직이거나 에이전트 자신이 동적

으로 모은 자료에 근거하여 이동할 수 있으며 온라인 쇼핑(online shopping)에서 분산된 과학적 수치계산에 대한 실시간 장치제어(real-time device control)에 이르기까지 폭넓게 쓰인다.

이동객체 시스템은 기존의 RPC(Remote Procedure Call)에 비해 몇 가지 장점을 가지고 있는데[3], 데이터의 처리를 직접 옮김으로써 네트워크 통신을 줄이고, 에이전트가 네트워크 상에서 작업을 하는 동안 그 클라이언트는 오프라인(off-line)상태가 될 수 있으며, 네트워크에 연결된 장치들을 실시간으로 제어할 수 있는 점을 대표적으로 들 수 있다.

2.2 기본적 요구사항



[그림 1] 이동 에이전트 시스템의 일반적인 구조

이동 에이전트를 위한 시스템은 일반적으로 그림 1과 같은 구조를 가지며 기본적으로 다음과 같은 특징을 갖추어야 한다[4]. 프로그래머가 에이전트 기반의 응용 프로그램을 구현하는데 있어 에이전트의 생성과 전송, 그리고 재생 등을 프로그래밍하기 쉬워야 하고, 에이전트는 각각의 이름으로 구분되고 처리된다. 에이전트끼리 상호 통신이 가능하고 동기화를 맞출 수 있으며 그 행동은 필요시 감독이 가능하고 전반에 걸친 제어가 가능해야 한다. 보안 측면에서는 에이전트가 가진 기밀 사항들이 보호되고, 인증(authentication)이나 접근 허용에 관한 메커니즘이 제공되어야 한다. 이를 지원하는 시스템은 이동객체를 지원하는 내부 구조를 가지고 있고 이동 에이전트를 위한 시스템 구조와 에이전트 서버 기능을 할 수 있어야 하고 에이전트를 전송할 수 있어야 한다.

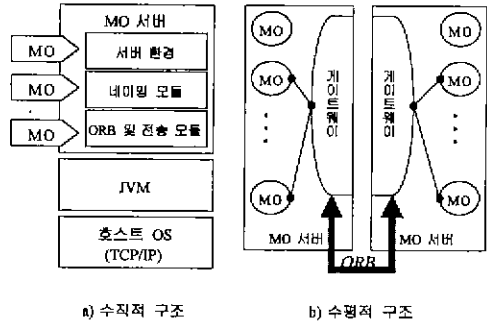
3. 이동객체 ORB 시스템 구성

3.1 시스템 개요

이동객체는 이동 에이전트의 좀더 일반적인 개념으로 이동객체 전형(mobile agent paradigm)을 제공하기 위해 사용되는 기본적인 요소이다. 이동객체는 메소드(method)와 데이터를 캡슐화하여 분산 시스템상의 노드들을 움직이고 이러한 이동객체를 위한 ORB 시스템은 이동 에이전트 모델을 구축할 수 있는 구조를 제공한다.

이러한 전형을 Java가 가진 객체지향성, 분산성, 보안성, 독립성, 멀티쓰레딩 지원 등의 특징을 이용하여 JVM상에서 구현한다. 이 시스템은 이동객체의 이동성을 지원하며 이동객체들 간의 통신에 대한 기능이 뛰어나고 네이밍 서비스(naming service)와 이벤트 서비스(event service)를 지원함으로써 객체 자체의 독립성이 강하고, 일반

Java 객체를 수용함으로써 확장성이 뛰어나다. 시스템의 개략적인 구성은 그림 2와 같다.



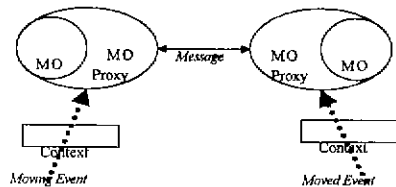
[그림 2] 이동객체 시스템의 구조

3.2 이동객체 ORB 시스템의 기능

이동객체 ORB 시스템은 이동객체를 처리하기 위한 기본적인 기능들을 에이전트 기술과 병합한 플랫폼으로 객체에 대한 동기화와 기타 제어를 담당하며 각 객체에 대해 여러 서비스를 제공한다.

1) 리모트 객체 요청(Remote Object Request)

다른 호스트의 객체에 대해 객체 생성이나 서비스와 같은 요청을 하면 그 객체에 대한 동일한 인터페이스(interface)의 프락시(proxy)가 리모트 객체로 반환된다. 시스템은 실시간에 동적으로 프락시를 만들고 이런 프락시를 통하여 실제 객체와의 메시지 교환 또는 서비스 요청과 결과 반송 등을 한다.



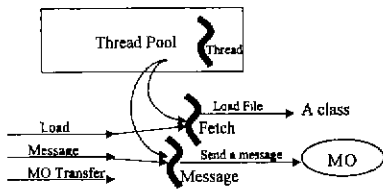
[그림 3] 리모트 객체의 생성

2) 메시지 전송

메시지 전송은 동기전송, 단방향 전송, 그리고 비동기 전송 모드를 지원한다. 이동객체가 이동 중일 때는 전달자(forwarder)를 생성하여 남겨두어 자신으로 오는 모든 메시지를 받을 수 있다. 각 이동객체는 메시지 큐(message queue)를 관리하고, 각 메시지는 우선 순위(priority)를 가진다.

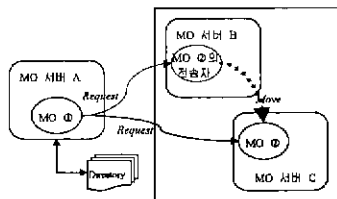
3) 객체 이동(Mobility Service)

이동하고자 하는 대상 서버에 대한 URI(Uniform Resource Identifier)로 이동할수, moveTo()를 간단히 호출함으로써 그 객체 자신이 다른 호스트로 이동할 수 있다. 실제 이동은 이동 관련된 메시지를 보내고, 대상 시스템에서 클래스를 로딩하고, 지정된 기능을 수행한다.



[그림 4] 객체 이동시의 스레드 사용

이동시 이 객체에 대한 전송자를 원 호스트에 남겨 둘으로써 이동 중에도 메시지를 넘겨받을 수 있다. 이동된 객체로 메시지를 보낸 객체는 그 결과에 새로운 위치정보를 결과로 받으므로 다음 요청시 전송자를 거치지 않고 바로 새 위치의 객체로 메시지를 전송한다.



[그림 5] 객체의 이동과 전송자

4) 에이전트 생성 및 관리

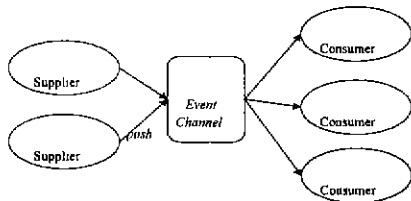
에이전트를 만들기 위해 기본적으로 이동성 등을 지원하도록 미리 만들어 놓은 Agent라는 클래스(class)를 확장하고, 이동 일정, 라이프사이클(lifecycle)등과 같은 여러 속성은 선택적으로 설정할 수 있다.

5) 네이밍 서비스(Naming Service)

객체에 관련된 모든 작업은 객체의 이름으로 참조가 되므로 객체에 대한 위치에 대한 투명성을 제공하고자 객체 위치와 고유 ID 등의 정보를 가진 참조 테이블(lookup table)을 관리하여 네이밍 서비스를 제공한다. 테이블에는 객체의 이름과 현재 참조 가능한 위치에 대한 URI, 그리고 기타 라이프사이클 정보 등을 관리한다.

6) 이벤트 서비스(Event Service)

이벤트 관심이 있는 이벤트에 대한 이벤트 채널(event channel)을 관리함으로써 비동기적인 객체 처리를 할 수 있다. 이는 OMG의 푸쉬 모델(push model)을 채택하였다.



[그림 6] 푸쉬형의 이벤트 서비스

7) 보안

Java가 제공하는 보안 메커니즘을 이용하며, 보안 관리자(security manager)를 선택적으로 설정할 수 있다.

3.5 인터페이스

시스템은 기본요소에 대한 API를 제공함으로써 그 앵스트렉션(abstraction)을 제공하므로 간단히 프로그램을 할 수가 있다. 이동 객체 시스템의 객체들은 다음과 같은 인터페이스들 중 그 객체가 가진 특성에 맞게 상속하여 생성된다.

[표 1] 기본적인 객체 인터페이스

인터페이스	설명
Context	이동객체의 실시간 환경에 대한 인터페이스
Message	객체간 또는 객체와 서버간의 메시지에 대한 인터페이스
Messenger	프락시를 통해 대상을 지정하고 실제적으로 원하는 메시지를 보내는 메신저에 대한 인터페이스
Mobility	이동성을 가진 객체에 대한 인터페이스로 이동과 전송자에 관련된 기능이 정의되어 있다.
Proxy	실제 리모트 객체에 대한 가상 참조를 제공하는 프락시에 대한 인터페이스로, 프락시의 생성과 프락시에 대한 메시지의 설정 등의 메소드를 선언한다
Agent	이동성과 기타 인터페이스를 집합적으로 가지고 이동 에이전트의 특성을 제공한다.

4. 결론

본 연구에서는 이동 에이전트 시스템의 특성에 맞는 이동객체를 설계하고, Java를 이용하여 이를 지원하는 ORB 시스템을 구현하였다. 이 시스템은 객체의 이동성을 제공하며 객체들간의 요청과 객체들 간의 통신을 지원하며 이러한 기능에 대해 간단하고 명료한 인터페이스를 제공한다. 에이전트 시스템이 갖추어야 하는 특징을 그대로 제공하므로 이 시스템을 이용하여 이동 에이전트 시스템을 쉽게 구축할 수 있다. 또한 ORB로서 요구되는 기본적인 서비스 즉, 네이밍 서비스, 이벤트 서비스를 제공하며, 다른 객체 모델을 수용할 수 있다.

앞으로 현재 적용한 Java의 기본 보안 메커니즘 외에 별도의 보안 시스템을 적용하여 보안을 강화하고, 트레이딩 서비스, 에이전트들끼리의 연동 서비스, 객체의 인증 서비스 등을 추가함으로써 좀더 안정된 시스템을 기대할 수 있고, CORBA나 DCOM와 같은 기존 ORB의 객체들과의 연동작업과 웹 적용 등의 시스템 확장을 연구하고자 한다.

5. 참고문헌

- [1] D.S. Mitijicis, S.Guday, F. Reynolds, D. Bolinger and P. Dale, "Mobile Objects and Agents", USENIX COOTS '98, April 1998.
- [2] J. Kinary and D. Zimmerman, "A Hands-on Look at Java Mobile Agents," *IEEE Internet Computing*, pp. 21-30, July 1997.
- [3] R. S. Gray. "Agent Tcl: A flexible and secure mobile-agent system," *In proceedings of the 4th Annual Tcl/Tk Workshop(TCL '96)*, July 1996.
- [4] N. M. Karnik and A. R. Tripathi, "System support for Mobile Agents," *Mobile Object Systems ECCOP Workshop '97*, May 1997.
- [5] D.S. Milijicis, S.Guday and R. Wheeler, "Old Wine in New Bottles: Applying OS Process Migration Technology to Mobile Agents", *Mobile Object Systems ECCOP Workshop '97*, May 1997.