

분산객체 환경에서 자바 기반의 이동 에이전트 시스템 설계 및 구현

°박정기, 정광수

광운대학교 전자공학부 컴퓨터통신연구실

Design and Implementation of Java Based Mobile Agent System in Distributed Object Environments

°Chunggee Park and Kwangsue Chung

Computer Comms. Lab., Kwangwoon Univ.

요 약

본 논문에서는 이동 에이전트가 작동할 수 있는 환경을 제공하는 이동 에이전트 시스템(Mobile Agent System)을 설계 및 구현하였다. 이동 에이전트 시스템은 자바 언어를 기반으로 하여 구현하였으며, OMG의 MAF(Mobile Agent Facility) 표준에 따라 구현하여 CORBA 분산환경에서의 상호운용성을 제공한다. 개발한 이동 에이전트 시스템을 이용하여 CORBA 기본 서비스 중에 하나인 명명 서비스를 실현하였다.

1. 서 론

인터넷의 급속한 보급 및 대중화는 인터넷에 연결된 컴퓨터 및 이를 사용하는 사용자의 수를 계속해서 엄청나게 증가시키고 있다. 이러한 시점에 기존의 클라이언트/서버 구조는 수많은 정보 요구자와 제공자의 연결을 처리하는데 있어서 그 한계를 드러냈고, 이를 해결하기 위한 방법으로 미들웨어 분산처리 기술이 발전하게 되었다. 객체지향 분산처리 기술의 표준인 CORBA는 이러한 분산처리 기술의 주류를 형성하게 되었다.

또한 최근 엄청난 양의 정보를 보다 효율적으로 관리 및 처리하기 위한 수단으로 이동 에이전트 기술이 대두되고 있다. 이동 에이전트는 사람을 대신해서 네트워크를 돌아다니면서 일을 자율적으로 처리해주는 프로그램을 말한다. 이동 에이전트 기술은 에이전트가 자율적인 이동 능력을 지님으로써 네트워크의 트래픽을 감소시킬 수 있고, 네트워크의 연결이 잠시 끊어졌을 때에도 지속적인 일의 수행이 가능하다는 장점 때문에 최근 들어 분산환경과 맞물려 정보 수집 및 검색, 네트워크 관리, 전자상거래 분야에서 널리 사용되고 있는 추세이다.

본 논문에서는 CORBA 분산환경을 바탕으로 자바언어를 사용하여 이동 에이전트 기술의 핵심인 이동 에이전트 시스템을 설계 및 구현하고, 이를 CORBA의 기본 서비스인 명명 서비스에 활용하는 방안을 제시하고자 한다.

2. 이동 에이전트 시스템의 설계

2.1 에이전트 시스템의 기본 개념

CORBA 표준을 주도하는 비영리 단체인 OMG(Object Management Group)에서는 분산환경 하에서 여러 가지 다양한 에이전트 시스템 간의 상호운용성을 제공하는 것을 목적으로 MAF(Mobile Agent Facility)를 CORBA 표준안에 포함시켰다. MAF를 참조하여 에이전트 시스템을 구성하는 기본적인 주요 요소들을 나열하면 다음과 같다

(1) Agent

사람을 대신하여 자율적으로 일을 처리하는 컴퓨터 프로그램

(2) Stationary Agent

자신이 기동된 시스템에서만 일을 수행할 수 있는 에이전트

(3) Mobile Agent

자신이 기동된 시스템을 벗어나서도 일을 수행할 수 있는 에이전트. 즉, 네트워크를 돌아다니며 일을 처리할 수 있는 이동성을 지닌 에이전트를 말한다.

(4) Agent State

에이전트가 일을 수행하고 이동할 때 다른 시스템에서 그 일을 다시 재개하기 위하여 에이전트에게 필요한 모든 속성 정보

(5) Agent Authority

에이전트를 기동시킨 사람 또는 조직을 지칭하는 일종의 저작권. 이것은 에이전트 인증 시에 필요하다.

(6) Agent Name

에이전트의 작동을 관리하기 위하여 에이전트를 식별하는데 요구되는 전역적으로 유일한 이름

(7) Agent System

에이전트를 생성, 해석, 실행, 전송, 종료시킬 수 있는 플랫폼

(8) Place

에이전트를 실행시킬 수 있는 에이전트 시스템이 가지고 있는 context로서 에이전트가 이동할 때 상주하는 실행 환경.

(9) Agent Location

에이전트 시스템에 있는 플레이스의 주소

(10) Region

같은 authority를 갖는 에이전트 시스템들의 집합

2.2 이동 에이전트 시스템의 구성

본 논문에서 구현한 이동 에이전트 시스템의 구성은 그림 1과 같이 크게 6개의 모듈로서 구성되어 있다.

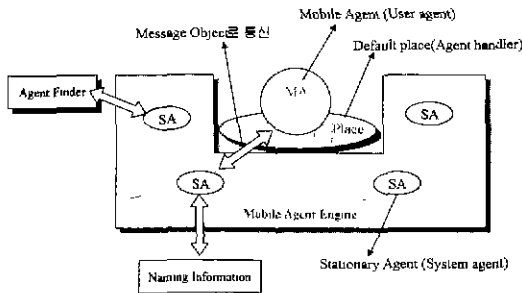


그림 1. 이동 에이전트 시스템(MOAS)의 구성

• 에이전트 모듈(Agent Module)

Agent의 기본 메소드를 추상 클래스로 정의하고, Stationary Agent인 System Agent와 사용자가 정의한 Mobile Agent인 User Agent가 공통적으로 상속받아 플레이스에 등록되어 실행될 수 있도록 해 준다. Agent 추상 클래스는 그림 2에서 보는 것과 같이 정의되어 있다.

```

abstract public class Agent
{
    abstract public void initialize();
    abstract public void start(),
    abstract public void stop(),
    abstract public void conclude();
}
    
```

그림 2 Agent 추상 클래스

• 통신 모듈(Communication module)

Mobile Agent와 Stationary Agent의 통신 수단을 제공하는 모듈로서 MessageObj 클래스를 사용하여 메시지 형태로 에이전트간에 정보를 주고 받는다

• 플레이스 모듈(Place Module)

플레이스는 Thread로 된 Agent Handler로 구성되어 Agent의 등록 및 관리의 역할을 한다.

• 라이프사이클 모듈(Lifecycle Module)

AgentLoader 클래스와 AgentServer 클래스로 이루어져있으며 등록된 에이전트에 대한 초기화, 실행, 멈춤, 종료 등의 생명주기를 관리한다. 그림 3은 Agent의 라이프사이클을 그림으로 보여주고 있다.

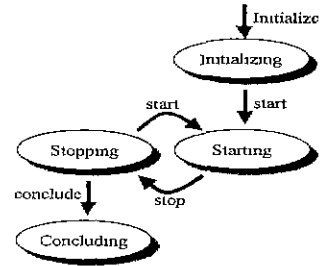


그림 3. Agent의 라이프사이클

• 계산 모듈(Computational Module)

에이전트 시스템의 중추로서 AgentEngine 클래스로 되어 있어 에이전트에 대한 생성, 등록, 조회, 전송 등의 역할을 담당한다.

• 네이밍 모듈(Naming Module)

분산환경에서 초기에 에이전트 시스템에 대한 객체 참조를 얻어내는 역할을 담당한다 즉, 에이전트 시스템에 대한 명명 서비스 모듈이라고 할 수 있으며 AgentFinder 클래스로 되어 있다

3. 이동 에이전트 시스템의 구현

이동 에이전트 시스템을 구현하는 데에는 Telescript, Agent Tcl 등 여러 가지 프로그래밍 언어가 사용되고 있지만 최근에는 자바 언어가 이동 에이전트 시스템 구현을 위한 Mobile Code로서 각광을 받고 있다. 자바 언어는 플랫폼에 독립적인 bytecode를 생성할 뿐만 아니라 멀티 쓰레드, Object Serialization, RMI(Remote Method Invocation)와 같은 이동 에이전트 구현에 필요한 핵심 기능들을 언어 자체에서 충실하게 지원하고 있다는 장점이 있다.

본 논문에서 구현한 이동 에이전트 시스템 역시 JDK 1.1.6 버전의 자바 언어로 구현하였으며, 이동 에이전트 클래스 파일이 Object Serialization을 통하여 자체 실행 코드와 상태 정보 코드가 다른 에이전트 시스템으로 전송된다. 전송된 객체는 자바 클래스로더에 의해서 인스턴스화 되어 실행되는 구조를 가지고 있다 한편, 이동 에이전트 시스템이 분산환경에서 상호운용성을 가지고 연동될 수 있도록 OMG MAF에 준하여 구현하였다

CORBA 환경을 지원하기 위하여 IONA사의 OrbixWeb 3.0을 사용하여 AgentEngine과 AgentFinder 클래스를 구현하였다. 구현된 이동 에이전트 시스템을 활용하기 위한 명명 서비스는 Orbix 2.1을 사용하여 C++-언어로 구현한 명명 서비스를 사용하였다 본 이동 에이전트 시스템 개발을 위하여 사용된 환경은 LAN으로 묶여진 펜티엄 200MHz급의 MS-Windows 시스템에서 구현 및 테스트하였다.

4. 동작 시나리오: 명명 서비스

앞으로 CORBA 분산환경이 대중화되면 가장 기본 서비스인 명명 서비스를 제공하는 수많은 네임 서버가 생겨나게 될 것이다 또한 각 네임 서버가 가지고 있는 네이밍 정보는 더욱 복잡하게 연결되어 있을 것이다. 이러한 환경에서 명명 서비스의 네이밍 정보 관리의 중요한 의미를 지닌다. 즉, 클라이언트의 요구에 의해 명명 서비스를 처리하기 위하여 계속적으로 다른 네임 서버에게 의뢰하여 최종적으로 명명 서비스를 제공하는 것은 비효율적이다. 가능하면 네임 서버가 로컬에 주변 네임 서버의 네이밍 정보를 캐싱하여 가지고 있으면 보다 빠르고, 네트워크 상황이 좋지 않은 상태에서도 안정적으로 명명 서비스를 제공할 수 있게 될 것이다. 이 점에 착안하여 이동 에이전트를 명명 서비스의 네이밍 정보 수집 및 관리에 활용해 보았다.

구현된 이동 에이전트 시스템을 활용하는 것을 간단히 보여주기 위하여 명명 서비스에서 네이밍 정보를 자율적으로 수집, 관리하는 이동 에이전트 응용 시나리오를 제작하였다. 그림 4는 이동 에이전트가 네이밍 정보를 수집하여 갱신하는 동작 시나리오를 그림으로 나타내 준 것이다. 이동 에이전트의 동작 메커니즘을 정리해 보면 다음과 같다.

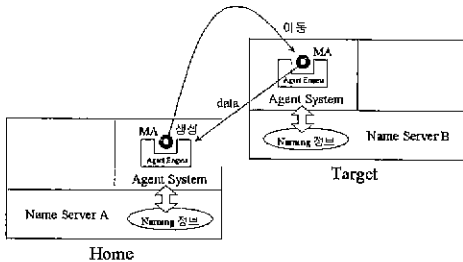


그림 4 이동 에이전트의 동작 시나리오

- (1) 홈 에이전트 시스템에서 먼저 이동 에이전트를 생성하여 등록하고, 이동 에이전트를 기동시킨다.
- (2) 기동된 이동 에이전트는 먼저 홈 네임 서버의 Stationary Agent를 통하여 네이밍 정보를 읽어들이어 검사하여 그림 5와 같은 네임 서버의 계층적 구조(Naming federation)가 있는 링크를 찾아내어 방문하고자 하는 다른 네임 서버의 목록(Itinerary)을 만든다

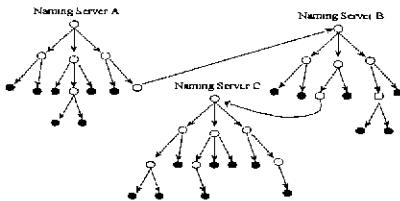


그림 5. 네임 서버의 계층적 구조

- (3) 목록이 만들어지면 지정된 시간에 이동 에이전트는 AgentFinder에 의해서 얻어진 객체 참조를 바탕으로 대상 에이전트 시스템으로 이동한다.

- (4) 이동된 에이전트는 대상 에이전트 시스템에 도착하여 인증, 등록 및 활성화가 되고, 네이밍 정보 수집 작업을 수행한다.
- (5) 네이밍 정보 수집은 이동 에이전트가 대상 시스템에 있는 네임 서버의 자원에 접근할 수 있는 Stationary Agent에게 네이밍 정보를 요구하게 되고, Stationary Agent는 네이밍 정보를 읽어들이어 메시지 객체를 통하여 이동 에이전트에게 네이밍 정보를 전달한다.
- (6) 이동 에이전트는 수집된 정보를 홈 에이전트 시스템에 전달하고 자신은 소멸된다
- (7) 홈 에이전트 시스템은 이동 에이전트로부터 받은 정보로 네임 서버의 네이밍 정보를 갱신한다

5. 결론 및 향후 연구과제

본 논문에서는 자바 언어로 이동 에이전트 시스템을 설계 및 구현하여 CORBA의 기본 서비스인 명명 서비스에 응용하였다. 이동 에이전트를 이용하여 네이밍 정보를 수집, 관리하여 명명 서비스의 성능을 개선시켜 보았다. 즉, 다른 네임 서버의 네이밍 정보를 이동 에이전트를 활용하여 수집하여 홈 네임 서버의 로컬에 캐싱해 놓음으로써 클라이언트가 명명 서비스를 이용하고자 할 경우, 로컬의 네이밍 정보를 이용하여 객체 참조(Object Reference)를 되돌려 줄 수 있으므로 속도 및 안정성에 있어서 향상된 효과를 거둘 수 있다. 또한 네임 서버가 이동 에이전트를 이용하게 되면 이동 에이전트가 자발적으로 정보를 수집해서 갱신해 주기 때문에 네이밍 정보 관리가 용이해지는 효과를 거둘 수 있다.

향후 연구과제로는 이동 에이전트 시스템을 실제로 활용하기 위하여 보안문제에 관한 연구가 필요하다.

참 고 문 헌

- [1] James E. White, *Mobile Agent White Paper*, General Magic Co., <http://www.genmagic.com/technology/techwhitepaper.html>
- [2] OMG, *The Common Object Request Broker. Architecture and Specification*, Rev 2.2, Object Management Group Inc., 1998
- [3] OMG. *CORBA Services: Common Object Services Specification*, Object Management Group Inc., 1997
- [4] Crystaliz Inc, General Magic Inc, GMD FOKUS and IBM Co., *Mobile Agent Facility Specification*. Joint Submission, OMG TC Document. 1997
- [5] Joseph P. Bigus. *Constructing intelligent Agents with Java*, John Wiley & Sons Inc., 1998
- [6] Danny B. Lange, *Aglet WorkBench Homepage*, IBM Co <http://www.trl.ibm.co.jp/aglets>
- [7] Joaclum Baumann, *Project Mole Homepage*, Stuttgart Univ., <http://www.informatik.uni-stuttgart.de/ivpr/vs/projekte/mole.html>
- [8] *Voyager Homepage*, ObjectSpace Inc, <http://www.objectspace.com/voyager/>