

# Jini 기반의 이동 에이전트 시스템

구형서<sup>o</sup>, 윤형석, 김진홍, 유양우, 문남두, 이명준  
울산대학교 컴퓨터정보통신공학부  
{masker, miracle, avenue, soft, dooya, mjlee}@mail.ulsan.ac.kr

## Jini-based Mobile Agent System

Hyeong-Seo Koo<sup>o</sup>, Hyeong-Seok Yoon, Jin-Hong Kim, Yang-Woo Yu, Nam-Doo Moon,  
Myung-Joon Lee  
School of Computer Engineering and Information Technology, University of Ulsan

### 요 약

Jini는 분산 컴퓨팅 환경에서 네트워크 플러그 앤 워크(Network Plug and Work)를 가능하게 해주는 기로서, 실세계의 네트워크에 연결된 주변기기나 응용프로그램을 Jini 서비스로 등록하여 동적으로 작업을 구성할 수 있는 새로운 분산 패러다임으로 자리잡고 있다.

본 논문에서는 Jini 기반의 이동 에이전트 시스템을 설계하고 구현하였다. SMART, Grasshopper, Aglets 같은 CORBA 기반의 이동 에이전트 시스템은 표준화, 동적 서비스 등록 및 에이전트의 영속성을 지원하기에 미흡하였으나, 개발된 시스템은 Jini 서비스를 이용하여 이러한 문제들을 체계적으로 해결하고자 하였다. 본 시스템은 Jini 서비스로 동작하는 이동 에이전트 시스템, 에이전트와 에이전트 시스템의 위치 정보를 관리하는 위치정보제공자, 그리고 이동 불가능한 에이전트를 임시 저장하는 이동경로관리자로 구성된다.

### 1. 서론

인터넷 기반 기술의 발달 및 네트워크 중심 프로그램의 수요 증가와 더불어 최근 이동 에이전트 시스템과 분산 응용 프로그램에 대한 관심이 증가하고 있다. 이동 에이전트 시스템은 분산 컴퓨팅 환경에서 이기종 시스템간의 상호운용성을 지원하여 이동 에이전트가 작업을 수행할 수 있는 환경을 제공한다. 이동 에이전트 기술을 이용하면 네트워크 부하를 줄이고, 사용자의 요구에 맞는 에이전트를 개발할 수 있을 뿐만 아니라, 불안정한 네트워크 환경에서도 안정적으로 수행할 수 있다. SMART[1], Grasshopper[2], Aglets[3] 등 이동 에이전트 기술을 이용하여 개발된 CORBA 기반의 이동 에이전트 시스템이다. 이동 에이전트는 분산 환경에서 전자상거래의 실시간 디바이스 컨트롤, 네트워크 관리, 그리고 과학적인 계산을 요구하는 분산처리와 같은 다양한 분야에 적용될 수 있다. 하지만, 이런 가능성에도 불구하고 현재 이동 에이전트 시스템은 표준화 및 응용프로그램의 부족 등으로 인해 널리 이용되지 못하고 있다.

한편, Jini 기술은 실세계의 네트워크에 연결된 주변기기나 응용 프로그램을 Jini 서비스로 등록함으로써 네트워크 플러그 앤 워크를 가능하게 하는 분산 패러다임으로 새롭게 자리잡고 있다.[4]

Jini를 기반으로 이동 에이전트 시스템을 개발하면 기능성 기반의 룩업서비스(lookup service)를 제공하여 원격 에이전트 시스템의 비스 객체의 프락시(proxy)를 이용함으로써 서로 다른 에이전트 시

스템에 의해 생성된 이동 에이전트와의 협력이 가능하며, 비연결성과 지역성을 지원함으로써 네트워크 부하를 줄일 수 있다.[5] 또한 Jini 서비스는 객체의 상태를 저장하고 관리할 수 있는 효율적인 매커니즘으로 JavaSpace를 제공한다.[6] 본 시스템에서는 에이전트의 저장소로 JavaSpace를 이용하였다.

본 논문에서는 Jini 서비스로 동작하는 이동 에이전트 시스템을 설계하고 구현함으로써 Jini 룩업 서비스를 이용한 동적인 이동 에이전트 시스템 등록이 가능하며, JavaSpace에 이동 에이전트를 저장하여 영속성을 지원하였다. 본 시스템은 Jini 서비스로 동작하는 이동 에이전트 시스템과 이동 에이전트와 이동 에이전트 시스템의 위치 정보를 관리하는 위치정보제공자, 그리고 이동 불가능한 이동 에이전트를 임시 저장하여 목적지 에이전트 시스템으로 이동시키는 이동경로관리자로 구성된다.

본 논문의 구성은 다음과 같다. 1장의 서론에 이어 2장에서는 Jini 기반의 이동 에이전트 시스템에 대해 기술하고, 마지막으로 3장에서는 결론 및 향후 연구과제를 기술한다.

### 2 Jini 기반의 이동 에이전트 시스템

Jini는 네트워크에 연결된 모든 장치나 응용프로그램을 Jini 서비스로 등록하여 사용한다. 그림 1은 Jini 기반의 이동 에이전트 시스템의 구조를 나타낸다. 본 시스템은 이동 에이전트 시스템, 위치정보제공자, 그리고 이동경로관리자로 구성되며, 이들은 모두 Jini 비스로 등록되어 사용된다.

† 본 연구는 정보통신부의 "정보통신우수시범학교지원사업"의 지원에 의해 이루어졌음.

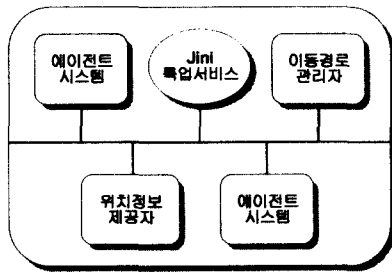


그림 1 시스템 구조

2.1 이동 에이전트 시스템

이동 에이전트 시스템은 이동 에이전트가 실행되는 공간인 플레이스와 이동 에이전트를 추적하고 감시하는 모니터로 구성된다.

2.1.1 플레이스

플레이스는 이동 에이전트가 실행되는 환경이다. 따라서, 플레이스는 데몬(daemon) 형태로 독립적으로 수행되어야 하며 플레이스 내의 에이전트들을 실행시킬 수 있어야 한다. 이러한 기능을 효과적으로 지원하기 위하여 플레이스는 병행성을 가진 쓰레드로 구현되었으며, 에이전트 시스템 내에서 여러 개의 플레이스가 독립적으로 동작할 수 있게 되었다.

플레이스는 에이전트를 저장하기 위한 데이터 구조로 ArrayList을 사용하여 에이전트의 이름과 속성을 저장한다. 저장된 에이전트는 플레이스에서 작업을 수행하게 된다.

2.1.2 모니터

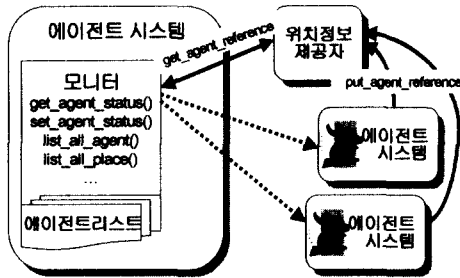


그림 2 모니터 구조

그림 2의 모니터는 플레이스와 이동 에이전트의 상태를 관리하는 모듈로서 이동 에이전트 시스템마다 존재하며 에이전트 시스템이 기동됨과 동시에 하나의 데몬 형태로 수행을 시작한다.

모니터는 현재 에이전트 시스템 내에서 동작 중인 플레이스 및 이동 에이전트의 이름과 개수를 실시간으로 저장하고 관리한다. 또한 모니터는 자신의 에이전트 시스템에서 실행 중인 이동 에이전트의 실행 상태와, 원격의 에이전트 시스템에 보내어 실행 중인 이동 에이전트의 상태를 감시한다. 원격의 에이전트 시스템에서 실행 중인 이동 에이전트의 감시는 위치정보제공자와 원격 에이전트 시스템의 모니터와의 상호작용으로 이루어진다.

모니터는 에이전트의 상태를 감시하는 기능과 더불어 에이전트의 실행 상태를 제어하는 기능이 있다. 예를 들어, 에이전트가 원격 에

이전트 시스템에서 수행 도중 시스템의 문제나 다른 요인으로 인해 그들의 수행이 중단될 수 있다. 이때, 사용자는 위치정보제공자와 모니터의 상호작용으로 이동 에이전트의 상태를 실시간으로 파악함으로써 이동 에이전트를 재실행시키거나, 종료시킬 수 있다.

2.1.3 이동 에이전트

이동 에이전트는 한 에이전트 시스템에서 다른 에이전트 시스템으로 이동하여 작업을 수행하는 프로그램이다. [7] 본 시스템에서 이동 에이전트는 자바 쓰레드로 동작하며, 미리 설정된 경로를 따라 이동하여 작업을 수행한다.

· 이동 에이전트의 생성 : 이동 에이전트는 이름, 에이전트 객체 플레이스, 소유자, 코드베이스, 그리고 클래스 등을 인자로 생성된다. 이동 에이전트의 이름은 네트워크에서 유일하며 이동 에이전트를 구별하는 식별자로 사용된다. 이동 에이전트 시스템은 로컬에서 이동 에이전트를 생성하고 실행시킬 뿐만 아니라, 원격의 에이전트 시스템에서도 에이전트를 생성하고 실행시킨다. 즉, 원격의 에이전트 시스템에서 이동 에이전트가 생성되어 실행하더라도, 생성 시 주어지는 인자를 통해 에이전트에게 필요한 코드를 실시간으로 전송하여 작업을 수행할 수 있다. 이동 에이전트는 네트워크를 이동하면서 위치정보제공자에 등록된 자신의 위치를 실시간으로 변경한다.

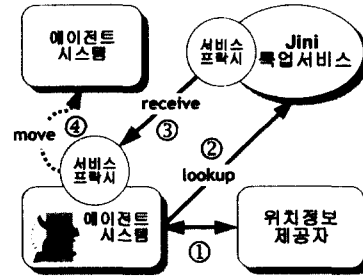


그림 3 에이전트의 이동

· 이동 에이전트의 이동 : 그림 3은 이동 에이전트의 이동과정을 보여준다. 이동할 목적지 에이전트 시스템이 결정된 이동 에이전트는 네트워크를 통해 전송될 수 있는 상태로 직렬화(serialization)다. 위치정보제공자를 통해 이동할 목적지 에이전트 시스템의 위치를 알아내고, Jini 룩업서비스에 등록된 목적지 에이전트 시스템의 서비스 프락시를 다운로드 받은 후 RMI(remote method invocation)를 통해 목적지 에이전트 시스템으로 전송이 이루어진다. 목적지 에이전트 시스템에 도착한 이동 에이전트는 역직렬화(deserialization)고, 해당 플레이스의 에이전트 저장소에 대기하다가 서버의 자원을 할당받아 작업을 수행한다.

· 이동 에이전트의 생명주기 : 그림 4와 같이 이동 에이전트 객체는 생성 직후 또는 저장소에서 꺼내는 다음 초기화된다. 초기화는 에이전트가 살아 있으나 실행되지 않은 상태이며, 클라이언트가 이동 에이전트가 수행할 작업을 조정하는 시기이다. 이동 에이전트 시스템에 도착한 이동 에이전트는 인증과정을 거친 후 에이전트 저장소에서 대기하다가 서버의 자원을 할당받아 실행한다. 이동 에이전트는 이동명령에 의해 작업을 중단하고 로컬 시스템에서의 작업을 마친 후, 다른 에이전트 시스템으로 이동하거나, 이동하지 못할 경

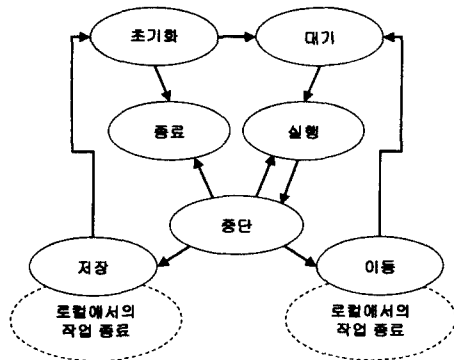


그림 4 에이전트의 생명주기

우 지속적인 저장소에 저장된다. 이동 에이전트는 작업수행 도중 에이전트를 생성한 소유자의 복귀명령을 받은 즉시 클라이언트로 되돌아간다. 클라이언트는 복귀한 이동 에이전트가 수집해온 자료를 뽑아내고 종료시키거나, 이동 에이전트를 초기화하여 새로운 임무를 부여한다.

2.2 위치정보제공자

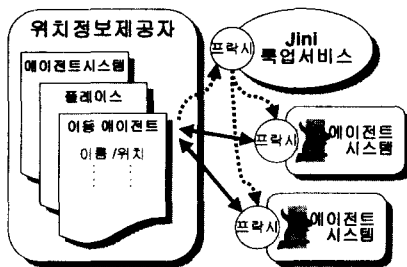


그림 5 위치정보제공자

위치정보제공자는 수행 중인 이동 에이전트 시스템과 플레이스, 그리고 이동 에이전트의 위치정보를 관리한다. 이동 에이전트의 위치란 이동 에이전트가 수행 중인 플레이스와 이동 에이전트 시스템의 이름을 말한다. 이동 에이전트와 이동 에이전트 시스템은 네트워크에서 유일한 이름을 갖는다.

사용자가 자신의 이동 에이전트의 위치를 알고자 하는 경우, 위치정보제공자에게 에이전트의 이름을 인자로 요청을 보내면 위치정보제공자는 이동 에이전트가 수행 중인 에이전트 시스템의 주소와 동작 중인 플레이스의 이름을 리턴해 준다. 마찬가지로 이동 에이전트가 목적지 에이전트 시스템으로 이동하고자 할 때에도 위치정보제공자를 통해 목적지 에이전트 시스템의 위치와 기동 중인 플레이스의 이름을 얻은 후 록업서비스에 있는 목적지 에이전트 시스템의 서비스 프락시를 다운로드 받아 RMI를 통해 이동한다.

2.3 이동경로관리자

이동경로관리자는 이동 에이전트를 임시로 저장하는 공간을 제공한다. 그림 6-①에서와 같이 이동 에이전트가 목적지 에이전트 시

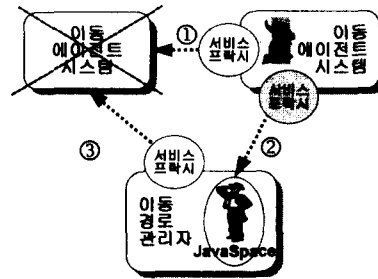


그림 6 이동경로관리자

스템이나 네트워크의 문제로 이동할 수 없을 경우, 그림 6-②와 같이 이동경로관리자의 JavaSpace에 임시로 저장된다. 이동 에이전트는 net.jini.core.entry.Entry 타입으로 저장되며 이동 에이전트 시스템의 이름, 목적지 에이전트 시스템의 이름, 그리고 이동 에이전트 객체를 인자로 가진다. 이동경로관리자는 그림 6의 ③과 같이 isAlive()를 이용하여 목적지 에이전트 시스템으로 접속이 가능한지를 일정 시간 간격으로 알아보고, 목적지 에이전트 시스템과의 접속이 가능한 경우 이동 에이전트를 목적지 에이전트 시스템으로 전송한다.

3 결론 및 향후 연구과제

본 논문에서는 Jini를 기반으로 이동 에이전트 시스템을 설계하고 구현하였다. 개발된 시스템은 이동 에이전트 시스템을 Jini 서비스로 등록하여 동적으로 작업환경을 구성할 수 있다. 이동 불가능한 에이전트는 이동경로관리자의 JavaSpace에 저장하여 영속성을 지원하였다. 또한 위치정보제공자에 이동 에이전트의 위치를 등록하여 실시간으로 에이전트의 위치와 상태를 감시하고, 실행을 제어할 수 있도록 하였다.

향후, 개발된 이동 에이전트 시스템을 보완하여 이동 에이전트간의 통신과 이동 에이전트의 보안을 지원할 계획이다.

참고문헌

- [1] 유양우, 김진홍, 구형서, 박양수, 이명재, 이명준, "SMART: OMG의 MAF 명세를 지원하는 CORBA 기반의 이동 에이전트 시스템", 한국정보처리학회 논문지 제8-C권 제2호, pp221~233, 2001.
- [2] S.Choy, T.Magedanz, "Grasshopper Technical Overview", IKV++ GmbH, 1999.
- [3] M.Oshima, G.Karjoth, K.Ono, "Agllets Specification 1.1 Draft", <http://www.trl.ibm.com/agllets/spec11.html>
- [4] Jan Newmarch, "Jan Newmarch's Guide to JINI Technologies", <http://jan.netcomp.monash.edu.au/java/jini/tutorial/Jini.xml>
- [5] Tao Tao, "JATWING: A Distributed Enhanced Tuple Space-based Mobile Agent Computation Model In A Ubicomp World.", University of Alabama at Birmingham, 2001.
- [6] Danny Ayers and etc, "Professional Java Server Programming", Wrox Press Ltd, 1999.
- [7] Neeran M. Karnik and Anand R. Tripathi, "Design Issues in Mobile-Agent Programming Systems", IEEE Concurrency, 1998.