논•문

Jini 기반의 이동에이전트 지원 시스템
(A Jini-based Mobile Agent Support System)

김 진 홍, 구 형 서, 안 건 태, 이 명 준
울산대학교 컴퓨터 · 정보통신공학부 전산학전공
{avenue, masker, java2u}@mail.ulsan.ac.kr
mjlee@uou.ulsan.ac.kr

요 약

이동에이전트 시스템은 에이전트를 대상 시스템으로 이동시켜 실행함으로써 네트워크 트래픽을 줄이고, 서버의 기본 기능을 변경하지 않고 사용자의 다양한 요구사항을 처리하는 프로그래밍 기법을 제공한다. Jini 기반 구조는 분산 응용을 위하여 간단하면서도 유연한 프로그래밍 기법과 네트워크 환경을 제공하고 있으며, 이를 통하여 이동에이전트 시스템의 동적인 등록 및 소재 파악의 기능과 에이전트의 활동에 유용한 서비스들의 동적 제공이 용이하게 지원될 수 있다. 본 논문에서는 Jini 기반 이동에이전트 지원 시스템인 JMAS 시스템에 관하여 기술하고 있다. JMAS 시스템은 이동에이전트 시스템의 기본 기능인 에이전트 생성, 관리, 전송, 위치 파악 및 에이전트간의 통신 기능을 제공하고 있으며, 나아가 이동에이전트 시스템의 신뢰성을 위하여 예외 상황 처리 및 이동에이전트 시스템의 영속성 지원 기능을 제공하고 있다.

1. 서론

인터넷의 사용이 보편화됨에 따라 빠르게 변화하는 사용자의 요구를 만족하면서도 편리하고 유용한 서비스를 제공하기 위하여 효과적인 분산 기술 및 분산 구조가 제시되어 왔으며, 그러한 노력의 일환으로서 사용자를 대신하여 자발적으로 행동하는 에이전트를 다른 시스템으로 이동시켜 작업을 수행하는 *이동에이전트 프로그래밍 패러다임*[1,2]과 분산 응용프로그램을 위하여 *튜플스페이스*[3] 프로그래밍 기법에 기반을 둔 객체공간 및 유연한 네트워크 환경을 제공하는 *Jini* 기술[4]이 주목을 받고 있다.

이동에이전트 프로그래밍 패러다임은 많은 양의 데이터를 주고받는 서비스를 제공할 때 에이전트를 대상 시스템으로 이동시켜 실행함으로써 네트워크 트래픽을 줄이고, 서버의 기본 기능을 변경하지 않고 사용자의 다양한 요구사항을 지원할 수 있다. 이동에이전트 프로그래밍 패러다임을 이

†본 연구는 정보통신부의 정보통신 우수시범학교 지원사업의 지원으로 수행되었음.

용한 서비스는 고급화된 전자상거래 서비스, 전자 경매, 네트워크 관리, 과학적인 계산을 지원하는 분산처리 서비스 등과 같이 다양하다.

이동에이전트 프로그래밍 패러다임을 이용한 시스템에 대한 연구는 산업단체 또는 연구단체에서 최근 활발하게 진행되어 왔지만, 통일된 분산 기반구조를 채택하지 못하여 널리 사용되지 못하고 있다. 상이한 기반구조에서 작동하는 이동에이전트 지원 시스템은 자신의 서비스를 사용자에게 편리하게 제공할 수 없으며, 사용자 또한 이동에이전트 지원 시스템의 서비스를 용이하게 활용하기 힘들다.

1999년 1월에 발표된 Jini 기술은 실생활에서 사용하는 주변기기나 응용프로그램을 쉽게 작성할 수 있는 프로그래밍 기법과 유연한 사용을 위한 환경을 제공하는 새로운 분산 패러다임으로 자리 잡고 있다.

Jini의 Lookup & Discovery 서비스[5]는 Jini 서비스 형태인 이동에이전트 시스템의 동적인 등록 및 검색, 에이전트의 이동, 에이전트가 이용할 서비스의 동적인 등록 및 검색 등을 기본적으로 제공할 수 있으며, Jini의 여타 기능도 이동에이전트 시스템의 개발을 용이하게 하여준다.

본 논문에서는 이러한 Jini 기반 구조에서 동작하는 이동에이전트 지원 시스템인 *JMAS(Jini based Mobile Agent support System)*에 대하여 기술하였다. 개발된 JMAS 시스템은 이동에이전트 시스템의 기능과 에이전트의 위치를 찾는 기능을 Jini 서비스 형태로 제공하여 이동에이전트 시스템의 서비스가 Jini 네트워크 하에서 Jini 프로그래밍 기법을 이용하여 자연스럽게 운용될 수 있도록 하였다. 그리고 신뢰성 있는 이동에이전트 서비스를 위하여 JMAS 시스템은 에이전트의 실행 시 발생하는 예외 상황의 처리 기능, 이동에이전트 시스템의 영속성(Persistence) 기능, 그리고 *JavaSpace*를 [6] 이용한 이동에이전트 간의 통신 기능을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 Jini 기반 구조와 최근 이동에이전트 지원 시스템 구현 경향에 대하여 살펴보고, 3장에서는 JMAS 시스템의 구조 및 각 모듈의 역할에 대해 설명한다. 4장은 이동에이전트 시스템의 신뢰성 지원을 위한 이동에이전트 시스템의 영속성 기능, 예외 상황 처리 기능, 그리고 에이전트간의 통신 기능에 대하여 기술한다. 5장에서는 JMAS 시스템의 실행에 대하여 살펴보고, 끝으로 6장에서는 결론 및 추후 연구 방향에 대하여 살펴본다.

2. 관련연구

2.1. Jini 기반 분산 기술

Jini 기술은[4] 네트워크 상의 지능형 기기들이나 소프트웨어들이 Jini 네트워크에 접속과 동시에 서비스할 수 있는 기능(Network Plug and Work)을 제공하고, 사용자의 요구에 의한 서비스 요청(Services on Demand)을 처리할 수 있는 새로운 분산 프로그래밍 패러다임을 제공한다. Jini 기반 구조(Infrastructure)는 Jini 서비스를 자바의 객체로 저장하여 사용자의 요구에 따른 서비스를 관리하는 서비스 관리자(Lookup Service), 사용자 및 서비스 제공자가 서비스 관리자의 위치를 자동적으로 파악하기 위한 Discovery 프로토콜, 그리고 서비스 제공자가 서비스 관리자에게 서비스를 등록하기 위한 Join 프로토콜로 구성된다. 그리고 분산 응용 프로그램의 개발을 위한 리스(Leases), 이벤트(Events), 그리고 트랜잭션(Transacti-

ons) 처리 프로그래밍 모델을 제공하여 분산된 자원의 효율적인 관리와 서비스간의 이벤트 및 트랜잭션 처리를 할 수 있다. 튜플스페이스(Tuple Sapce)는 분산된 객체들에 대한 효율적인 관리를 용이하게 제공할 수 있는 프로그래밍 패러다임을 제공한다. 이 튜플스페이스를 기반으로 개발된 JavaSpace는 분산 객체의 영속성과 데이터 교환 기능을 지원하는 Jini 서비스로써, 표준 인터페이스를 통하여 객체를 저장하고 가져오는 기능 및 탐색 기능을 지원한다.

2.2. 타 이동에이전트 시스템 분석

본 장에서는 JMAS 시스템과 관련이 있는 Aglets[7], Concordia[8], Ajanta[9], 그리고 SMART[10,11,12] 시스템에 대하여 간략하게 살펴본다.

IBM의 Aglets은 "weak" 이동성[13]을 지원하는 Java 기반의 이동에이전트 시스템이다. Aglets은 call-back 기반의 프로그래밍 모델을 이용하여 에이전트에게 이벤트가 발생할 때마다 에이전트의 특정 메쏘드를 호출할 수 있다. Aglets은 자신의 위치와 에이전트의 위치를 식별하기 위해 호스트 이름과 포트번호를 조합한 DNS 기반의 URL을 이용한다.

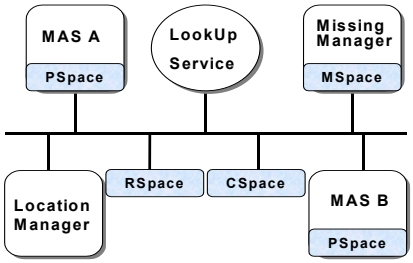
Concordia는 Java를 이용하여 개발된 이동에이전트 시스템으로써, Java의 직렬화와 클래스로딩 매커니즘을 이용하여 에이전트의 이동을 지원하지만, 쓰레드 레벨의 실행 상태 저장은 지원하지 않는다. Concordia에서는 신뢰성 있는 에이전트 전송과 서버나 에이전트의 복구 기능은 객체의 영속성 매커니즘을 적용하여 제공되고 있다. Concordia에서는 서버의 자원과 에이전트를 보호하기 위하여 사용자 ID를 이용하고, 악의 사용자로부터 에이전트와 시스템을 제한하기 위하여 암호화 프로토콜을 사용한다.

Ajanta는 Java 직렬화 매커니즘을 이용하여 에이전트의 이동성을 지원하는 이동에이전트 시스템이다. Ajanta 시스템에서 에이전트 코드는 에이전트가 명시한 서버로부터 동적으로 다운로드되어 사용될 수 있다. 그리고, 공개키 프로토콜을 적용한 암호화 및 사용자 인증기능을 제공함으로써 에이전트 코드와 상태를 악의의 호스트나 에이전트로부터 방어할 수 있다. 에이전트의 이동 경로를 프로그래밍하기 위하여 이주 패턴(migration pattern)의 개념을 사용하였으며, 에이전트 이동성은 "weak" 속성을 지원한다.

SMART는 Java 언어를 이용하여 OMG MAF 명세를 따라 구현된 시스템으로 이기종간의 상호운용성을 지원하는 CORBA 기반의 이동에이전트 시스템이다. OMG MAF 명세에서 제시하고 있는 표준 인터페이스를 통하여 에이전트를 생성하고, 전송하고, 실행시키는 기능을 지원하고 있으며, 에이전트 및 이동에이전트 시스템의 위치를 파악하는 기능을 제공하고 있다. 또한 이동에이전트 시스템의 자원 접근 및 보안 정책을 지원한다.

3. JMAS의 구조 및 기본 기능

JMAS 시스템은 MAS(Mobile Agent System), 위치관리자(Location Manager), 이동관리자(Moving Manager), RSpace(Resource Space), 그리고 CSpace(Communication Space)로 구성된다. (그림 1)은 Jini 서비스로 동작하는 JMAS의 각 구성요소를 포함한 구조를 나타내고 있다.



(그림 1) JMAS 시스템의 구조

3.1. 이동에이전트

3.1.1 이동에이전트

JMAS 시스템에서 이동에이전트의 속성은 (표 1)에서 보여주고 있다. 이동에이전트는 Java의 직렬화(Serialization)를 통하여 네트워크를 통하여 전송될 수 있는 객체로 선언된다. 그리고 이동에이전트이름은 이동에이전트의 식별자로써 생성시간, 생성한 에이전트 시스템 이름, 작성자로 구성된다.

<표 1> 이동에이전트의 속성

속 성	의 미
Name	이동에이전트 이름(생성시간, 시스템 이름, 작성자)
Place	이동에이전트가 실행할 MAS 내의 플레이스 이름
Creator	이동에이전트를 생성한 사용자 이름(MAS의 이름)
Author	이동에이전트를 소유한 사용자 이름
Path	이동에이전트가 이동할 MAS의 경로
Resource Level	MAS의 자원을 접근할 접근 권한
Location	이동에이전트의 위치
run()	이동에이전트의 실행 루틴

3.1.2. 에이전트의 이동(Mobility)

JMAS 시스템은 에이전트의 상태 정보만을 전송하는 약한 이동성(weak mobility)을 제공한다.

다음과 같은 과정으로 JMAS의 에이전트는 목적지 MAS으로 이동한다.

① 이동에이전트는 목적지 MAS의 이름을 가지고 자신이 실행하고 있는 MAS에게 이동할 것을 요청한다.

② 요청을 받은 MAS은 이동에이전트를 직렬화(Serialization)하고, 위치관리자를 통하여 목적지 MAS의 이름으로 위치정보(Location)를 알아낸다.

③ 요청을 받은 MAS은 목적지 MAS의 위치 정보를 가지고 Jini LookUp Service를 이용하여 목적지 MAS의 서비스 프락시를 다운 받는다.

④ 목적지 MAS의 프락시를 통하여 목적지 MAS의 전송 함수를 호출하여 직렬화된 에이전트를 전송한다.

⑤ 목적지 MAS은 전송된 이동에이전트를 역직렬화(Deserialization)하고 지정된 플레이스에 저장하고 실행시킨다.

⑥ 위치관리자에 이동한 에이전트의 위치 정보를 변경한다.

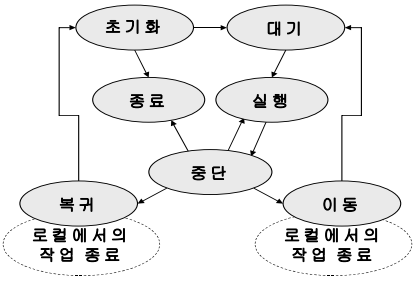
목적지 MAS는 이동에이전트의 계정과 권한 정보를 이용하여 에이전트를 인증하고, 인증된 이동에이전트는 목적지 MAS의 자원관리자를 통하여 시스템의 자원을 이용할 수 있다.

3.1.3. 에이전트의 생명주기(Life Cycle)

(그림 2)는 이동에이전트의 생명주기(상태변경)를 보여주고 있다.

- 초기화: 생성된 직후의 상태
- 대기: 플레이스의 에이전트 저장소에 저장된 상태

- 실행: 플레이스에서 실행 중인 상태
- 중단: 생성자 또는 실행 중인 MAS의 요구에 따라 실행에서 중단된 상태
- 종료: 이동에이전트의 종료 상태
- 이동: 직렬화되어 전송되는 상태
- 복귀: 생성된 MAS으로 이동되는 상태



(그림 2) 이동에이전트 생명주기

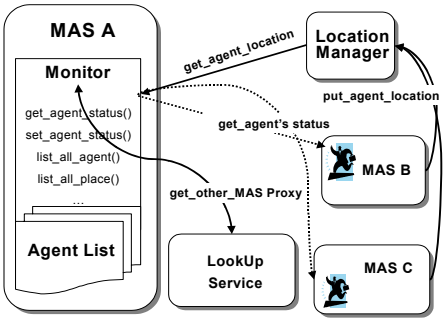
3.2. MAS의 모듈

3.2.1. 플레이스(Place)

플레이스는 이동에이전트의 실행환경을 제공한다. 플레이스는 에이전트 저장소를 제공하여 현재 자신에 수행되는 이동에이전트를 관리한다. 플레이스는 MAS에서 독립적으로 여러 개가 존재할 수 있으며, 실행될 플레이스가 지정되지 않은 이동에이전트는 기본적으로 존재하는 DefaultPlace에서 실행된다.

3.2.2. 모니터(Monitor)

모니터는 MAS의 플레이스 및 이동에이전트의 상태를 검색하고 제어하는 기능을 제공하며, 이동한 에이전트에 대한 원거리 제어를 할 수 있다. (그림 3)은 모니터가 플레이스 또는 이동에이전트의 상태를 알아보는 작동과정을 보여주고 있다. (그림 3)에서는 원거리의 에이전트 상태를 알아보기 위해서 위치관리자를 이용하여 에이전트가 실행 중인 MAS의 모니터를 이용한다.

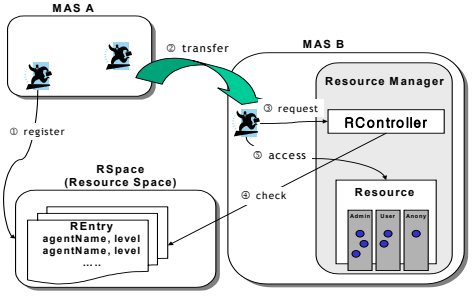


(그림 3) JMAS의 모니터의 동작

3.2.3.자원관리자(Resource Manager)

자원관리자는 이동에이전트가 MAS의 자원을 접근하여 사용할 때, 에이전트의 자원 접근 레벨에 따른 자원 할당여부를 결정하는 기능을 제공한다.

자원관리자는 MAS의 자원 접근을 세 가지 클래스인 Admin_Service, User_Service, 그리고 Anony_Service을 통하여 접근할 수 있도록 하고 있다. 각 클래스는 자원등급에 맞추어 에이전트가 수행할 수 있는 저 수준의 API를 정의하고 있다. (그림 4)는 자원관리자가 이동에이전트가 요구한 자원의 할당과정을 보여주고 있다. 자원 사용의 등급을 비교한 후, 자원을 할당한다. 할당받은 자원 접근 객체를 통하여 이동에이전트는 MAS의 자원을 이용하여 실행할 수 있다.



(그림 4) 자원관리자의 자원 할당 과정

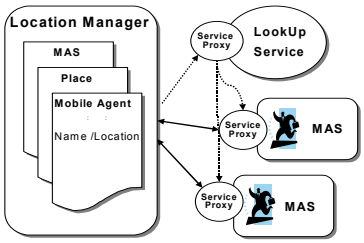
3.3. 위치관리자(Location Manager)

위치관리자는 MAS와 플레이스, 그리고 에이전트의 위치 파악을 위한 위치정보를 관리한다. 위치정보는 LookUp Service를 이용하여 실제 프락시를 얻을 때 사용된다.

위치정보는 문자열이며, JMAS으로 시작하고 MAS 이름과 플레이스 이름으로 구성된다. 다음은 JMAS 시스템에서 사용하는 위치정보를 나타내는 프로토콜이다.

jmas://mas-name:port/place-name/

(그림 5)은 위치관리자를 이용하여 다른 이동에이전트 시스템의 서비스 프락시를 얻는 과정을 보여주고 있다.

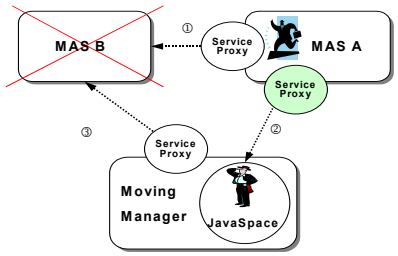


(그림 5) 위치관리자의 동작

3.4. 이동관리자(Moving Manager)

이동관리자는 목적지 시스템으로 이동하지 못하는 이동에이전트를 관리한다. 이동하지 못하는 에이전트는 MAS의 이름, 목적지 MAS의 이름, 그리고 이동 에이전트 객체를 정보를 MSpace에 저장된다.

(그림 6)에서 이동에이전트가 목적지 MAS으로 이동하지 못하는 경우를 처리하는 과정이다. 이동에이전트가 목적지 MAS의 문제나 네트워크의 문제로 이동할 수 없을 경우(그림 6-①), 이동에이전트는 이동관리자의 MSpace에 임시로 저장(그림 6-②)된다. 이동관리자는 isAlive()를 이용하여 목적지 MAS으로 접속이 가능한 지를 일정 시간 간격으로 알아보고, 목적지 MAS과의 접속이 가능한 경우 이동 에이전트를 목적지 MAS으로 전송한다.(그림 6-③)



(그림 6) 이동관리자의 작동 과정

3.5. 에이전트간의 통신

JMAS 시스템에서는 이동에이전트간의 통신 기능을 CSpace(Communication Space)이름의 JavaSpace를 이용하여 지원하고 있다. CSpace는 <표 2>에서 보여주는 인자를 가진 MEntry로 정의된 메시지를 저장한다. CSpace는 특정 메시지 저장될 때 지정된 형태의 메시지를 받을 Listener인 MAS로 메시지를 전달한다.

<표 2> 이동에이전트간의 메시지 형태

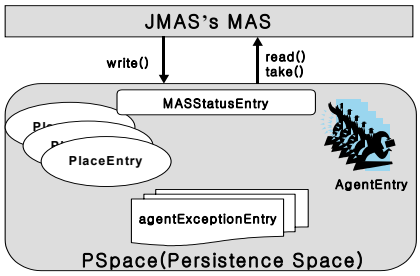
속 성		의 미
Sender_Agent		메시지를 보내는 에이전트 이름
Receiver_Agent		메시지를 받을 에이전트 이름(null: 그룹에 전달 시)
Agent_Creator		메시지를 전달받을 에이전트를 생성한 에이전트 시스템 이름
Agent_Author		메시지를 전달받을 에이전트를 소유하고 있는 사람 이름
Type	unique	특정 에이전트에게만 전달 방식 지정
	g_creator	Agent_Creator에서 기술된 에이전트 시스템이 만든 모든 에이전트에게 전달 방식 지정
	g_author	Agent_Author에서 기술된 에이전트 소유자의 모든 에이전트에게 전달 방식 지정
Message		전달될 메시지

4. 신뢰성을 위한 JMAS의 기능

JMAS에서는 이동에이전트 시스템의 상태 정보를 유지할 수 있는 영속성 기능, 그리고 에이전트 및 에이전트 시스템의 실행 시 발생 할 수 있는 예외를 처리할 수 있는 기능을 제공한다.

4.1. MAS의 영속성

JMAS 시스템은 MAS에서 실행되고 있는 이동에이전트, 플레이스, 그리고 시스템의 정보를 PSpace(Persistence Space)이름의 JavaSpace에 저장한다. MAS은 다시 기동될 때 PSpace에 저장된 정보를 가지고 이전 상태로 복구될 수 있다. (그림 7)은 PSpace에 저장되는 Entry의 종류 및 이를 다루는 오퍼레이션을 보여주고 있다.



(그림 7) MAS의 영속성 지원

정상 또는 비정상 종료를 고려한 MAS의 재 시작과정은 다음과 같다.

① MAS을 초기화.

② PSpace에 있는 플레이스 생성 및 실행.

③ PSpace에 있는 MASStatusEntry을 참조하여 시스템의 종료상태 판별.

- 정상 종료: PSpace의 모든 에이전트를 각각의 플레이스에서 실행시키다.
- 비정상 종료 시, 에이전트가 가진 자원 접근 등급에 따라 처리한다.(4.2.1절 참조)

4.2. JMAS의 예외처리 기능

JMAS 시스템은 MAS의 비정상적인 종료 때 발생하는 예외 상황과 함께 목적지 MAS으로 이동할 수 없거나, 이동한 목적지 MAS의 자원을 사용하지 못하는 세 가지 예외상황을 고려하여 처리한다. 발생된 예외 상황은 에이전트 시스템의 자원 접근 권한에 따라 처리되며, 발생된 예외 상황은 에이전트를 생성한 MAS에게 전달된다.

4.2.1. MAS의 비정상 종료 시 발생하는 예외 처리

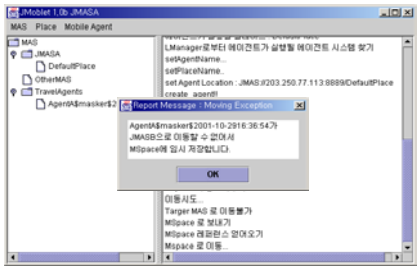
이동에이전트 시스템의 비정상 종료에 따른 예외 상황은 시스템이 재 시작하면서 이전의 상태를 복구할 때 시스템의 상태를 변경할 수 있는 권한을 가진 에이전트에서 발생된다. 이러한 경우, JMAS 시스템에서는 이동 에이전트를 생성한 MAS에게 예외를 통보하고 에이전트를 강제로 종료시킨다.

4.2.2. MAS의 자원 접근 시 발생하는 예외 처리

JMAS 시스템은 시스템의 정보나 상태를 변경하는 권한을 가진 에이전트가 수행 중 자원에 접근하지 못하는 경우, 이동에이전트 시스템은 에이전트를 만든 생성자에게 통보하고, 이전에 사용중인 자원을 반납하고 에이전트를 종료시킨다. 그러나 시스템의 자원에 영향을 미치지 않는 에이전트의 예외 상황의 경우, 에이전트를 생성한 MAS의 응답에 따라 재실행 또는 종료하도록 한다.

4.2.3. 이동에이전트의 이동 시 발생하는 예외 처리

이동에이전트가 목적지 MAS의 문제 혹은 네트워크의 문제로 인해 이동을 수행하지 못하는 경우, JMAS 시스템은 이동에이전트를 이동관리자의 JavaSpace에 임시로 저장해 두고 실시간으로 목적지 MAS으로의 이동가능 여부를 확인한 후 이동을 시도하도록 한다. 이동에이전트는 MissingAgentEntry 형태로 JavaSpace에 저장된다. (그림 8)은 이동에이전트가 MAS으로 이동하지 못할 때 발생하는 예외 상황을 생성한 MAS에게 보고한 모습을 보여주고 있다.

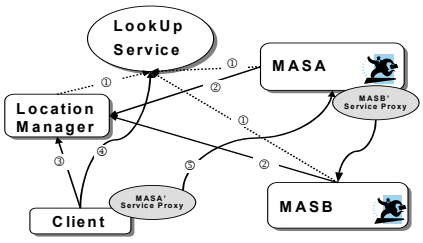


(그림 8) 이동불가 예외처리

5. JMAS 시스템 구현 및 실행

5.1. JMAS 시스템의 동작

사용자에 의하여 생성된 이동에이전트는 지정된 MAS으로 이동하여 자신의 작업을 수행한다. (그림 9)는 MAS의 기동 및 등록 과정과 생성된 이동에이전트가 이동하는 과정을 보여주고 있다.



(그림 9) 이동에이전트 이동 과정

1) LookUp Service와 위치관리자에 등록(MASA, MASB, 위치관리자).(그림 9-①②)

2) 사용자는 위치관리자에서 MASA의 이름으로 위치정보를 알아낸다.(그림 9-③)

3) LookUp Service에서 MASA의 프락시를 가져온다.(그림 9-④)

4) MASA의 프락시를 통하여 이동에이전트를 MASA에 전송한다.(그림 9-⑤)

5.2. 이동에이전트 프로그램 예

(그림 10)은 자신의 정보(agentCount)를 이동한 에이전트 시스템에서 증가시키는 간단한 에이전트 예제이다. (그림 10)에서 작성된 이동에이전트 프로그램은 이동에이전트 자신이 스스로 지정된 경로를 따라 이동하면서 에이전트 자신의 작업을 수행함을 보여주고 있다.

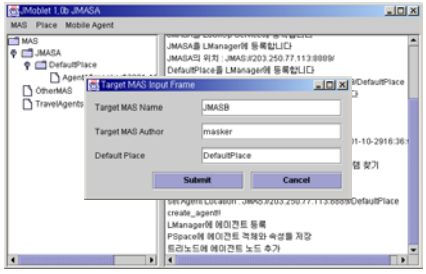
```
public class Agent implements Runnable, Serializable {
    public String agentName;
    private int agentCounter;
    private LinkedList agentPath;
    ...
    public Agent(String name, int counter, LinkedList
path) {
            this.agentName = name;
            this.agentCounter = counter;
            this.agentPath = path;
    }
    public void run() {
            // Change it's data
            for(int I = 0; I < 10; I++)
                agentCounter++;
            // Move to the next AgentStation
            if(agentPath.size() != 0) {
                next = (String) agentPath.get(host);
                agentPath.remove(host);
                gotoNextHost(next);
            }
}
```

(그림 10) 간단한 이동에이전트 프로그램

5.3. 실행 결과

JMAS 시스템은 이동에이전트의 생성과

이동 그리고 상태를 나타내는 기능을 위한 GUI를 제공하고 있다. MAS의 실행 모습과 기능을 (그림 11)에서 보여주고 있다. (그림 11)은 실행 중인 플레이스와 에이전트를 트리의 형태로 나타내고 있으며, MAS의 종료 및 재시작, 이동에이전트의 생성, 전송, 상태 변경의 에이전트 관리 기능과 플레이스의 생성, 삭제의 플레이스 관리 기능을 위한 툴바를 제공하고 있다. 또한 트리 패널에서 선택한 에이전트를 대상 MAS으로 이동하는 모습을 보여주고 있다.



(그림 11) MAS 실행 모습

6. 결론 및 향후 연구 방향

이동에이전트 프로그래밍 기법은 에이전트의 실행 코드와 관련 상태를 이동시켜 목적지 시스템에서 실행하는 기능을 통하여 인터넷 환경에서 특정 영역의 분산 애플리케이션을 개발하는데 유용하게 사용되고 있으며, 분산 서비스를 유연성 있게 지원하는 Jini 기술은 이동에이전트 시스템의 기본 기능을 구현하기에 적합한 분산 기반구조 및 프로그래밍 기법을 제공하고 있다. Jini에서 제공하는 Lookup & Discovery 서비스는 분산되어 있는 Jini 서비스 형태인 이동에이전트 시스템의 서비스를 동적으로 등록하고 등록된 서비스를 용이하게 찾을 수 있는 기능을 편리하게 제공할 수 있는 프로그래밍 기법을 제시하며, 또한 이동에이전트를 자연스럽게 목적지 이동에이전트 시스템으로 이동시키고 그 위치를 추적할 수 있는 기능을 지원한다.

본 논문에서는 이러한 Jini 기반 구조에서 제공하는 프로그래밍 기법을 사용하여 Jini 서비스로 동작하는 이동에이전트 지원 시스템인 JMAS에 대하여 기술하였다. JMAS 시스템은 에이전트 생성, 실행, 전송, 관리 등의 이동에이전트 시스템의 기능과 이동에이전트의 위치 파악 기능을 Jini 서비스 형태로 제공하고 있다.

JMAS의 MAS은 이동에이전트의 실행환경을 제공하며, 에이전트의 생명주기(Lifeycle), 플레이스의 실행, 그리고 시스템 자원을 제어하는 기능을 제공하고 있다. JMAS의 위치관리자는 이동에이전트 시스템 및 동적으로 변화하는 이동에이전트의 위치를 추적할 수 있는 기능을 제공한다. 또한, JMAS 시스템은 신뢰성 있는 서비스를 위하여 에이전트의 실행 시 발생하는 예외 상황 처리 기능, 이동에이전트 시스템의 영속성(Persistence) 기능, 그리고 이동에이전트 간의 통신 기능을 제공하고 있다.

현재의 JMAS 시스템은 JavaSpace의 보안 기능이 취약함으로 인하여 이동에이전트 및 이동에이전트 시스템간의 안전한 통신 방법을 제공하지 못하고 있다. 앞으로 보안적 기능을 제공하는 JavaSpace를 개발하여 에이전트간 통신을 안전하게 지원할 있는 보안 프로그래밍 기법을 개발 할 예정이며, 나아가 Jini의 Leasing 및 Transaction 프로그래밍 기법을 활용하여 이동에이전트의 작업관리를 지원하도록 시스템을 확장할 계획이다.

참고문헌

[1] N. M. Karnik and A. R. Tripathi, "Design Issues in Mobile-Agent Programming Systems", University of Minnesota, Dept. Computer Science and Engineering, IEEE, 1998.

[2] N. Carriero, D. Gelernter, J. Leichter, "Distributed Data Structures in Linda", proc. ACM Symp. Principles of Programming Languages, New Your:ACM, pp. 236-242, 1986.

[3] C.G. Harrison, D.M. Chess and A. Kershenbaum, "Mobile Agents: Are they a good idea?", Research Report, IBM, 1997.

[4] Jan Newmarch, "Jan Newmarch's Guide to JINI Technologies", http://jan.netcomp.monash.edu.au/java/jini/tutorial/Jini.xml.

[5] Scott Oaks and Henry Wong, "Jini in a Nutshell", Oreilly Press, March 2000.

[6] Danny Ayers and Hans Bergsten, "Professional Java Server Programming," Wrox Press Ltd, August 1999.

[7] IBM Aglets. Available at URL: http://www.trl.ibm.com/aglets/.

[8] Mitsubishi Concordia. Available at URL: http://www.meitca.com/HSL/Projects/Concordia/.

[9] Minnesota Ajanta. Available at URL: http://www.cs.umn.edu/Ajanta/.

[10] 유양우, 김진홍, 구형서, 박양수, 이명재, 이명준, "SMART: OMG의 MAF 명세를 지원하는 CORBA 기반의 이동에이전트 시스템", 한국정보처리학회 논문지, 제8-C권 제2호, 221-233, 2001.

[11] 유양우, 김진홍, 이명재, 박양수, 이명준, "SMART 이동 에이전트 시스템의 안전한 자원 접근 정책", 한국정보과학회 봄 학술발표논문집 Vol. 27, No. 1, p364-366, 2000.

[12] 유양우, 김진홍, 안건태, 문남두, 박양수, 이명준, "OMG MAF 명세를 지원하는 이동 에이전트시스템의 개발", 한국정보과학회 가을 학술발표논문집, Vol. 26, No. 2, p715-717, 1999.

[13] Alfonso Fuggetta, Gian Pietro Picco, Giovanni Vigna, "Understanding Code Mobility", IEEE Transaction On S/W Engineering, Vol. 24, NO. 5, May 1998.

김진홍

1999년 2월 울산대학교 전자계산학과 졸업(학사)

2001년 2월 - 울산대학교 컴퓨터정보통신 공학부 졸업(석사)

2001년 3월 - 현 울산대학교 컴퓨터정보통신 공학부 박사과정

관심분야 :이동에이전트 시스템, 생물정보학 등.

구형서

2001년 2월 울산대학교 컴퓨터·정보통신공학부 졸업(학사)

2001년 3월 ~ 현재 울산대학교 컴퓨터·정보통신공학부 공학석사과정

관심분야 : 이동에이전트 시스템.

안건태

1999년 2월 울산대학교 전자계산학과 졸업(학사)

2001년 2월 울산대학교 컴퓨터·정보통신공학부 졸업(석사)

2001년 3월 ~ 현재 울산대학교 컴퓨터·정보통신공학부 공학박사과정

관심분야 : 이동에이전트 시스템, 생물정보학.

이명준

1980년 2월 서울대학교 수학과 졸업(학사)

1982년 2월 한국과학기술원 전산학과 졸업(석사)

1991년 8월 한국과학기술원 전산학과 졸업(박사)

1982년 3월 - 현 울산대학교 컴퓨터정보통신 공학부(교수)

1993년 8월 - 1994년 7월 미국 버지니아대학 교환교수

관심분야 : 프로그래밍언어, 분산 객체 프로그래밍 시스템, 병행 실시간 컴퓨팅, 인터넷 프로그래밍시스템, 생물정보학 등.