

# 지니 홈네트워크상의 효율적인 서비스 관리 시스템 개발

정 준 영<sup>†</sup> · 정 민 수<sup>††</sup> · 김 광 수<sup>†††</sup>

## 요 약

지니는 자바 기술의 특성을 가장 잘 살린 홈네트워크 미들웨어이다. 하지만 지니를 기반으로 홈네트워크 서비스를 제대로 지원하기 위해서, 지니 시스템은 어려운 일련의 작업과 복잡한 관리를 필요로 한다. 본 논문에서 개발한 서비스 관리 시스템은 자동화 모듈을 포함하고 있는 서비스 제공자 컴포넌트와 룩업 서비스 컴포넌트이다. 본 자동화 모듈은 지니 시스템에서 필요한 라이브러리, 실행하고 있는 컴퓨터의 환경정보, 지니 서비스를 위한 클래스 파일 시스템 구성 정보의 탐색과 설정을 자동화하였다. 본 시스템은 지니 홈네트워크 실행환경의 자동화, 서비스 관리 구조의 단순화 및 서비스 실행의 시각화를 통해 서비스 관리자나 서비스 제공자에게 효율적인 서비스 관리 환경을 제공한다.

## Development of an efficient Service Management on Jini HomeNetwork

Jun-Young Jung<sup>†</sup> · Min-Soo Jung<sup>††</sup> · Kwang-Soo Kim<sup>†††</sup>

## ABSTRACT

Jini is a promising HomeNetworking middleware of computing environment based on Java Technology. To support HomeNetwork service based on Jini, Jini device requires a successive operation and complicated management. In this paper, our service management system is a service provider component and lookup service component including automation module. Our automation module provide searching and setting function of a library, runtime environment and class file system configuration information for Jini service. Our system can be accomplished by automation of runtime environment, simplification of service management structure, visualization of service execution.

키워드 : 지니(Jini), 홈네트워크(HomeNetwork), 서비스 관리(Service Management)

### 1. 서 론

인터넷 인구의 폭발적인 증가와 인터넷을 기반으로 하는 다양한 서비스와 함께 경제의 중심이 디지털과 인터넷으로 이동되고 있는 현시점에서 홈네트워크는 새로운 산업의 장르로 강조되고 있다. 현재 정보가전기기 및 디지털 기기들은 인터넷과 홈네트워크에 연결하기 위해서 지속적으로 개발되고 있다. 이와 더불어 가정 내의 홈네트워크 인프라는 여러 가지 종류의 네트워크 미디어와 프로토콜이 혼재하고, 정보가전 및 디지털 기기의 다양성으로 인해서 매우 복잡한 분산 네트워크 환경이 구축될 것이다[3, 15, 20].

홈네트워크의 복잡성과 이질성은 각종 정보가전 및 디지털 기기를 사용자의 개입없이 자동으로 구성하고 관리하는 기술을 요구한다[1, 3, 16, 17]. 또한 원격지에서 정보가전 및

디지털 기기들을 동작시킬 수 있는 서비스의 개발이 하부 홈네트워크의 물리 매체나 운영체제에 무관하고 용이한 환경을 제공할 수 있는 기술을 필요로 한다. 이러한 기술들 중 선택 마이크로 시스템즈사의 지니(Jini)와 마이크로소프트사의 유니버설 플러그 앤 플레이(UPnP: Universal Plug and Play)가 그 대표적이다[1, 4, 9, 16]. 네트워크상에 분산되어 있는 각종 장치들을 관리하기 위하여 지금까지 많은 연구개발이 진행되어 왔으나 이러한 시스템들은 확일적이고 유연성이 부족하다. 이에 반해 지니는 특별한 목적에 따라 만들어진 컴포넌트들을 모아서 관리 서비스를 재구성할 수 있는 기반구조를 가진다[1, 4]. 그리고 지니는 자바 기술 기반으로 자바 애플리케이션 환경을 네트워크 전체로 확장하여 모든 디지털 기기나 소프트웨어, 서비스를 연계할 수 있는 유용한 네트워크 시스템이며, 객체 기술이 사용되는 '객체 중심적인 분산 네트워크 아키텍처'이다[1, 4, 17, 18]. 하지만 서비스 제공자나 서비스 관리자는 시스템의 환경정보와 네트워크 관련 정보, 서비스 파일 시스템의 구조등을 사전에 파악해야 하고, 자바 환경이나 지니 시스템 환경을 잘 이해하고 있어야 한다. 지니 서비스를 제공하기 위해서 서비스 제공

\* 본 연구는 2003학년도 경남대학교 학술논문공개 연구비 지원으로 이루어 졌음.

† 준 회 원 : 경남대학교 대학원 컴퓨터공학과

†† 종 신 회 원 : 경남대학교 정보통신공학부 교수

††† 정 회 원 : 한국전기연구원 책임연구원

논문접수 : 2003년 1월 7일, 심사완료 : 2003년 5월 24일

자나 서비스 관리자는 실행방법에 관한 지식도 습득해야 하는데, 그 방법 또한 어렵고 매우 복잡하다.

본 논문에서 제안하고 있는 지니 서비스 관리 시스템(이하 JSMS : Jini Service Management System)은 지니 시스템을 이용하여 서비스를 제공하기 위한 방법을 자동화하고 서비스를 관리하는 구조를 단순화하여, 서비스 제공자나 서비스 관리자에게 효율적인 지니 시스템 환경을 제공하고자 한다.

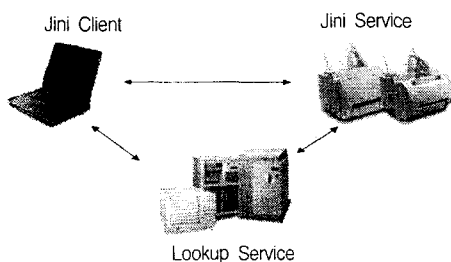
본 논문의 구성은 다음과 같다. 2장에서는 지니 시스템의 전반적인 구성요소와 서비스 아키텍처, 그리고 다른 홈네트워크 미들웨어에 대해서 간단히 기술한다. 3장에서는 지니 시스템의 보다 효율적이고 향상된 서비스 관리를 위해 필요한 사항들을 기반으로 JSMS가 가져야 할 기능들에 대해 설계하고, 4장에서는 3장에서 언급된 JSMS의 기능을 중점적으로 구체화하고 테스트하는 구현 논점에 대해서 기술한다. 그리고 5장에서는 본 논문에서 제안된 JSMS의 결론 및 활용 방향을 제시한다.

## 2. 관련 연구

### 2.1 지니 시스템

지니의 목적은 여러 장치들의 그룹들과 소프트웨어 컴포넌트들을 단일한 동적 분산 시스템으로 통합하는 것이다. 다양한 종류의 디바이스와 소프트웨어 자원의 연합체를 구성하여 서비스와 자원을 공유하고, 사용자의 위치변화에 상관없이 네트워크상의 플러그 앤 플레이(plug and play)기능을 가능하게 해주는 분산 환경 홈네트워크 자원 공유 플랫폼이다[1].

지니 시스템은 (그림 1)과 같이 세개의 구성요소로 이루어져 있다. 첫 번째는 프린터, 토스터와 같은 네트워크 이용자에게 문서 출력이나 빵을 만드는 서비스를 제공해주는 지니 서비스(Jini Service)이고, 두 번째는 지니 서비스를 사용하는 지니 클라이언트(Jini Client)이며, 세 번째는 지니 서비스와 지니 클라이언트를 서로 연결시켜 주는 룩업 서비스(Lookup Service)이다[1, 2, 4, 7]. 지니 시스템은 네트워크 프로토콜에 독립적이지만 현재는 TCP/IP 기반 네트워크로 연결되며, 실행코드들은 객체 직렬화(Object Serialization)를 이용하여 지니 서비스와 지니 클라이언트 그리고 룩업 서비스 사이를 이동할 수 있다. 코드의 이동은 자바 소켓에서



(그림 1) 지니 시스템 구성요소

지원하며, 자바 RMI(Remote Method Invocation : 원격 메소드 호출)를 이용하여 다른 자바가상기계 상에서 실행중인 객체의 메소드를 호출할 수 있다[4, 5, 17].

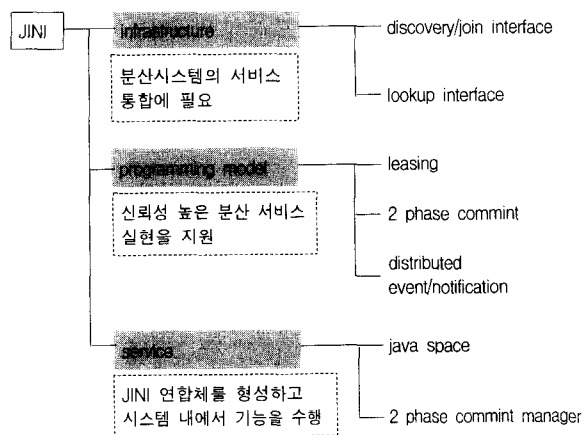
### 2.2 지니 컴포넌트

지니 시스템의 컴포넌트는 (그림 2)와 같이 세 가지로 분류할 수 있는데, 인프라스트럭처(Infrastructure), 프로그래밍 모델 그리고 서비스이다[1, 4, 18].

인프라스트럭처는 지니 시스템의 연합을 구성하는데 필요한 컴포넌트의 집합체로서 디스커버리(Discovery)/조인(Join)과 룩업(Lookup) 인터페이스로 이루어져 있다. 디스커버리/조인은 서비스를 검출하여 다른 멤버(서비스)에게 그 서비스의 존재를 통지하고 필요한 서비스를 제공 및 취득하며, 룩업 서비스를 찾는 기능을 수행한다. 룩업은 서비스(자바 프로그래밍 언어의 객체)의 저장장소로 기능하고, 그 관리와 업로드/다운로드를 담당하는 룩업 서비스를 수행한다[1, 4, 6, 12, 18].

프로그래밍 모델은 신뢰성 있는 서비스의 구성을 가능하게 하는 인터페이스의 집합으로 리싱(Leasing), 2-페이즈 커밋(2-Phase Commit), 분산 이벤트/통지(Distributed event/notification) 인터페이스로 구성된다. 리싱은 서비스에 대한 시간적인 액세스를 관리한다. 즉 시간제하에 따른 리소스의 할당과 해제를 수행한다. 2-페이즈 커밋은 트랜잭션 처리의 위탁과 수탁을 통해 지니 객체의 변경 및 관리를 처리한다. 분산 이벤트/통지는 자바빈즈와 표준 자바 이벤트 모델(Java Event Model)을 분산환경으로 확장한 개념이다[1, 4, 18].

서비스는 연합체를 형성하고 시스템내에서 기능을 수행하기 위한 연합체 내부의 엔티티로서 자바스페이스(JavaSpace)와 2-페이즈 커밋 매니저가 있다. 자바스페이스는 분산환경에 대한 상태보존(객체관리)를 담당하고, 2 페이즈 커밋 매니저는 트랜잭션의 위탁과 수탁을 관리한다[1, 4].



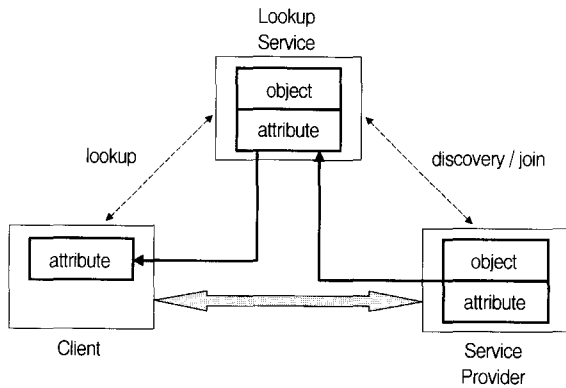
(그림 2) 지니 컴포넌트

### 2.3 지니 서비스 구조

지니 서비스 구조는 (그림 3)에서 보는 바와 같이 서비스

제공자(Service Provider), 룩업 서비스(Lookup Service) 그리고 클라이언트(Client)가 디스커버리, 조인 그리고 룩업 프로토콜을 이용하여 서로 통신하는 구조이다. 서비스 제공자와 클라이언트는 기본적으로 자바 RMI를 사용하여 상호작용 한다[1, 4, 5, 7, 8].

먼저 서비스 제공자가 지니 시스템에 플러그인(plug-in) 되면, 서비스는 룩업 서비스를 찾기 위하여 멀티 캐스팅 또는 유니 캐스팅을 이용한다[1, 18]. 룩업 서비스를 찾으려면 서비스 객체와 속성을 룩업 서비스에 등록한다. 클라이언트는 자바 타입과 서비스 속성을 룩업 서비스에게 요청하고, 서비스 객체의 복사본을 가져온다[1, 19]. 그러면 클라이언트는 서비스 객체를 통해 서비스 제공자와 상호작용 할 수 있다[4, 7].



(그림 3) 지니 서비스 구조

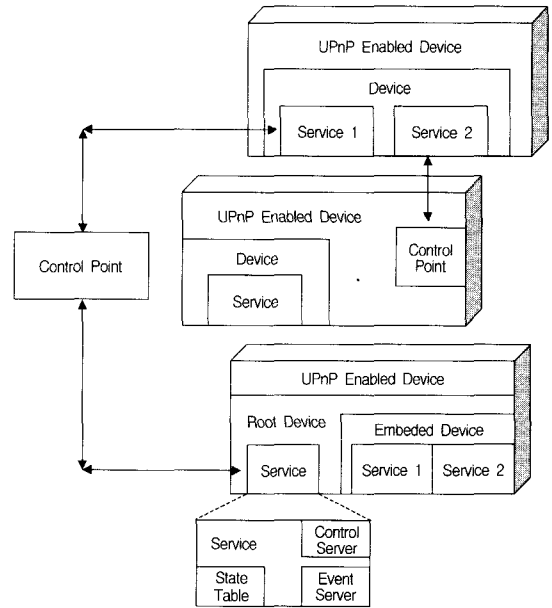
디스커버리는 룩업 서비스를 찾기 위한 프로토콜로서, 서비스 제공자나 클라이언트는 디스커버리 프로토콜의 멀티 캐스트나 유니캐스트 방법을 사용한다[1, 4, 18]. 그리고 서비스 제공자는 조인 프로토콜을 사용하여 룩업 서비스에 자신의 서비스 객체와 속성을 등록한다. 그러면 클라이언트는 룩업 프로토콜을 이용하여 룩업 서비스에게 등록되어 있는 서비스 객체나 속성을 검색하고, 원하는 속성에 해당하는 서비스 객체를 가져올 수 있다. 서비스 제공자와 클라이언트는 서로 RMI 통신이 가능해지며, 서비스 제공자와 클라이언트는 서로 통신하여 서비스를 실행하게 된다[1, 4, 8, 12].

2.4 UPnP

UPnP(Universal Plug and Play)는 홈네트워크 환경에서 운영체제, 언어 및 하드웨어에 독립적인 서비스 환경을 구축하기 위해 정보가전 및 디지털 기기 제어 소프트웨어 표준을 정의하고 있다[9]. UPnP 네트워크의 기본 블록은 (그림 4)에서 보는 바와 같이 디바이스와 서비스, 그리고 컨트롤 포인트이다.

UPnP 디바이스는 서비스의 컨테이너이다. 서비스는 UPnP 네트워크에서 가장 작은 컨트롤 단위로서, Action을 공개시키고 상태를 모델링한다. UPnP 디바이스의 서비스는 State Table, Control Server, Event Server로 구성된다. State

Table는 서비스의 상태를 모델링하고, 상태 변경시 업데이트한다. Control Server는 Action 요청(request)을 수신하면, Action을 실행하고, State Table을 업데이트한다. 그리고 응답(response)을 반환한다. Event Server는 서비스의 상태 변경시 네트워크에 접속되어 있는 가입자(subscriber)들에게 이벤트를 알린다.



(그림 4) UPnP Control Points, Devices and Services

Control Point는 다른 디바이스를 발견(discovering)하고 제어(controlling)할 수 있는 컨트롤러이다. 디스커버리가 일어나면, Control Point는 다음과 같은 작업의 수행이 가능하다.

- 디바이스 디스크립션(device description)을 검색하고 연관된 서비스의 리스트를 획득한다.
- 서비스에 대한 서비스 디스크립션을 검색한다.
- 서비스를 제어하기 위한 Action을 호출한다.
- 서비스의 Event Source에 서브스크라이브(subscribe)한다. 서비스의 상태 변경시 Event Server는 Control Point에게 이벤트를 송신할 것이다.

2.5 지니와 UPnP

지니는 기기간 통신을 위하여 기기에 지니 소프트웨어(자바 가상기계 기반)를 장착하는 것이 전제인데 반해, UPnP는 통신 프로토콜 레벨의 규정에 지나지 않아, IP를 이용할 수 있다면 가능하다. 그래서 UPnP는 IP를 기반으로 PnP(Plug and Play)를 네트워크와 피어-투-피어(peer-to-peer)로 확장하여 정보가전 및 디지털 기기, PC, 서비스가 투명하게 연동할 수 있는 개방형 표준을 정의하고 있으며, HAVi와 상호연동하기 위한 구조를 설계중이다. <표 1>은 지니와 UPnP의 기술을 비교하고 있다[4, 9, 17].

〈표 1〉 지니와 UPnP

|      | 목 표                                     | 기 술 내 용  |
|------|---|--|
| Jini | 홈네트워크 환경에 적합한 서비스 확산을 위한 하부 구조 정의       | <ul style="list-style-type: none"> <li>Jini는 IP 기반의 분산된 환경에서 적합하며, 소프트웨어 서비스에 적합한 기술로 구성됨</li> <li>차세대 정보가진 분야인 유비쿼터스(Ubiquitous) 컴퓨팅 환경으로 확장 가능한 기술임</li> </ul> |
| UPnP | 홈네트워크 환경에서 디바이스를 연결하고 제어를 가능하게 하는 기술 정의 | <ul style="list-style-type: none"> <li>인터넷에서 인증된 기술을 기반으로 홈네트워크에 적합한 기술 개발(디바이스는 TCP/IP와 IP가 필요함)</li> </ul>   |

UPnP를 구성하는 요소 기술은 홈네트워크에 연결되는 기기에 자동 IP 주소 할당 및 관리하는 제로 컨피규레이션(Zero-Configuration), 홈네트워크 상에서 디바이스나 서비스를 찾는 디스커버리 메커니즘(Discovery Mechanism), 도메인 네임 서버가 없는 환경에서 도메인 네임 서버 서비스를 상세히 기술하는 데이터 구조인 디바이스 디스크립션 스키마(Device Description Schema) 및 서비스 스펙(Service-specific) 또는 디바이스 스펙(Device-specific) API가 있다[9].

### 3. 지니 서비스 관리 시스템 : JSMS(Jini Service Management System)

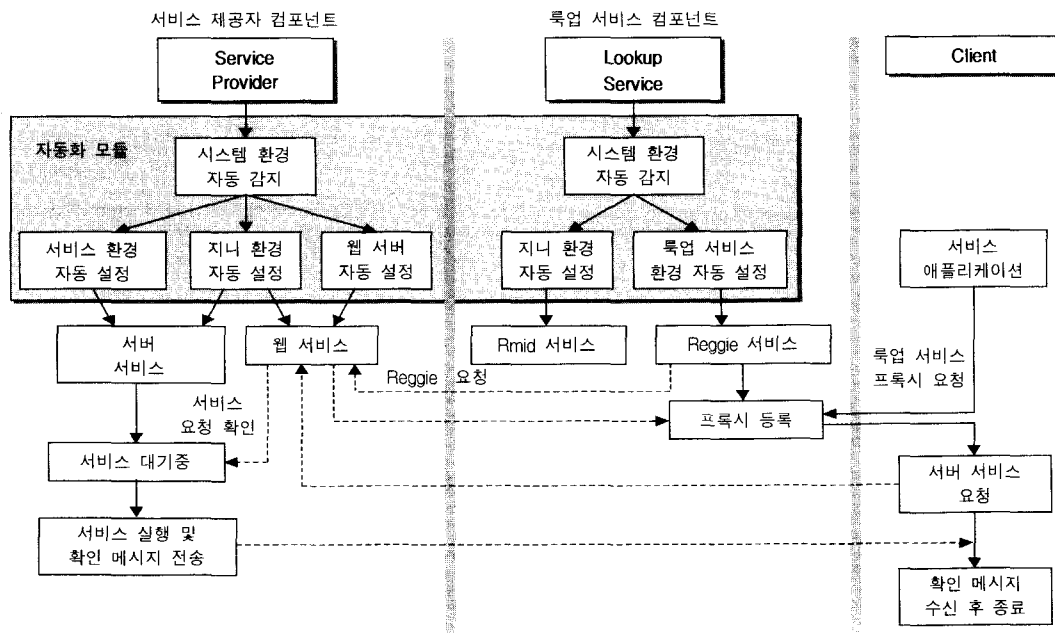
JSMS는 지니 서비스를 제공하기 위한 방법을 자동화하기 위해 자동화 모듈을 개발하여 사용하도록 하였으며, 서비스 관리가 용이하도록 하기 위해 일정한 서비스 관리 구조를 갖도록 한다. 또한 현재 진행되고 있는 서비스의 상태를 쉽게 파악하도록 서비스가 실행되는 과정을 시각화 시킴으로써 보다 유연한 서비스 제공이 가능하도록 하는데 그

주안점을 둔다.

#### 3.1 서비스 관리 시스템 구조 설계

지니 시스템의 효율적이고 일관성 있는 서비스 제공 및 관리를 위해서는 서비스 클래스 파일을 클라이언트보다는 서비스 제공자와 룩업 서비스에서 관리해야 한다. 지니에서 제공되는 모든 서비스는 자바 클래스 파일 형태로 제공되는데, 이는 지니가 자바 기반 객체지향 분산환경 시스템이기 때문이다.

JSMS는 지니 서비스 시스템이 본 논문에서 개발한 자동화 모듈을 포함하도록 하였다. 자동화 모듈은 실행되는 플랫폼에서 지니 서비스가 상호 유기적으로 수행되도록 하기 위해, 서비스 제공자와 룩업 서비스가 런타임에 필요한 지니 시스템 및 하드웨어적인 환경정보를 자동으로 획득하여 시스템에 적용한다. 자동화 모듈을 통해 설정된 각종 정보들은 서비스 제공자의 서비스 제공자 서비스와 웹 서비스를 실행하는데 사용된다. 그리고 룩업 서비스의 rmid 서비스와 reggie 서비스를 실행하는데 사용된다. 편사에서는 룩업 서비스로서 reggie를 제공하고 있으며, reggie를 수행하기 위해서는 rmid를 실행해야 한다. 이는 서비스 정보가 룩업 서비스에서 서비스 이용자(클라이언트)로 전송되는데 서비스 정보는 서비스 제공자와 통신을 위해 필요하며 rmid 서비스가 사용된다. 이렇게 서비스 제공자와 룩업 서비스가 실행되고 있는 지니 서비스 시스템에 클라이언트가 서비스를 받기 위해 접속하게 되면 서비스 제공자는 클라이언트에게 서비스를 제공하게 된다. 본 논문에서 제안하고 있는 JSMS를 적용하여 지니 서비스 시스템을 구성할 때 새로운 지니 서비스 구조는 (그림 5)와 같다.

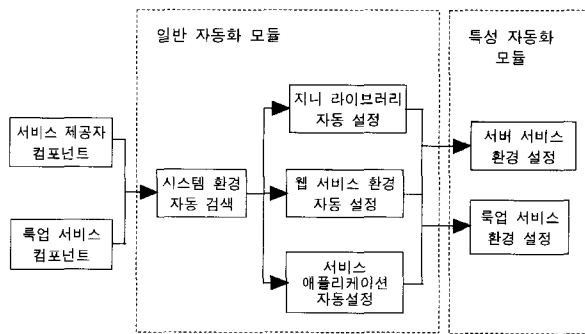


(그림 5) JSMS를 이용한 새로운 지니 서비스 구조

3.2 서비스 관리를 위한 자동화 모듈 설계

본 논문에서 설계한 지니 서비스 관리를 위한 자동화 모듈은 (그림 6)와 같이 크게 일반 자동화 모듈과 특성 자동화 모듈로 구성되어 있다.

일반 자동화 모듈은 서비스 제공자와 록업 서비스에서 공통적으로 적용이 되는 시스템의 초기화, 환경 설정 그리고 관리를 위한 모듈이다. 서비스 제공자나 록업 서비스가 실행되는 시스템의 환경 정보를 검색하여 수집한다. 그리고 검색 및 수집된 정보를 이용하여 지니 시스템을 구동시키는데 필요한 기본적인 환경을 스스로 설정한다. 또한 이 모듈은 지니 시스템에서 기본적으로 필요로 하는 라이브러리를 검색하여 지정하고, 클라이언트에게 서비스를 제공하는데 필요한 서비스 클래스 파일을 검색하여 위치정보를 알아낸다. 즉, 서비스 애플리케이션의 위치 정보를 검색하여 실행시에 클라이언트의 서비스 요청이 있을 때 이용 가능하게 환경을 설정한다. 서비스 제공자와 록업 서비스, 그리고 클라이언트가 상호 작용하기 위해 필요한 웹 서비스의 기본적인 환경을 자동적으로 검색설정하여 제공한다.



(그림 6) 지니 서비스 관리를 위한 JSMS 자동화 모듈

특성 자동화 모듈은 서비스 제공자와 록업 서비스의 고유한 서비스를 제공하기 위해 필요한 시스템의 초기화 및 환경 설정과 그 관리를 위한 모듈이다.

먼저 서비스 제공자는 두 가지 서비스를 제공하는데, 웹 서비스와 실제로 클라이언트가 이용하고자 하는 서버 서비스이다[1, 2, 4]. 먼저 웹 서비스는 TCP/IP 네트워크를 통해서 제공될 호스트 서버의 서비스를 위한 환경 설정 모듈이다. 서비스를 필요로 하는 클라이언트에게 다운로드 가능한 클래스 파일을 export 할 수 있도록 HTTP 서비스를 제공한다. 서버 서비스는 서비스가 제공될 클래스 파일의 위치 정보를 알아내어, 서비스가 제공될 클래스 파일을 관리하기 위한 디렉토리 환경으로의 위치정보를 재구성하는 것이다. 즉 서버 서비스가 제공되기 위한 클래스 파일들은 service 디렉토리에 위치하도록 하고, 서비스와 관련된 다운로드 가능한 클래스 파일들은 service-dl 디렉토리에 위치하도록 한다. 이는 본 시스템에서의 서비스 관리를 위한 구조로 다시 재구성하는 것이다. 그 다음 서버 제공자에서 제공될 서버

서비스 클래스 파일의 프록시 정보 파일을 록업 서비스와 클라이언트에게 익스포트(export) 가능하도록 한다. 이는 록업 서비스에 등록될 정보를 말한다[1, 4, 7, 8].

록업 서비스는 reggie 서비스 모듈과 로그 파일 설정 모듈이 있다. reggie 서비스 모듈은 서비스 제공자로부터 받은 프록시 클래스 파일을 획득하여 서비스 리스트에 등록시킨다. 그리고 여러 개의 서비스들을 관리하는 역할을 하는 reggie에 관련된 환경을 검색하여 자동적으로 설정한다. 이는 서비스 제공자에서 제공하는 서비스 관련 정보를 획득하기 위해 필요한 정보를 설정한다. 로그 파일 설정 모듈은 rmi 데몬 서비스와 reggie 서비스가 실행될 때 서비스와 관련된 로그 파일의 생성 및 저장 환경을 검색하여 자동적으로 설정한다.

3.3 서비스 파일 시스템 구조 설계

JSMS는 서비스 제공자와 록업 서비스의 서비스 관리를 위해 일정한 형태의 파일 시스템 구조를 가지도록 <표 2>와 같이 지니 서비스를 위한 클래스 파일의 위치 정보를 재구성한다. 이것은 일관된 서비스 제공 및 관리 구조를 구성함으로써, 서비스의 실체인 클래스 파일을 보다 손쉽게 유지 관리하기 위함이다.

<표 2> 지니 서비스 파일 시스템 디렉토리 구조

|                  | Directory Name | Description                      |
|------------------|----------------|----------------------------------|
| Jini Service Env | Client         | 클라이언트의 구현 클래스 파일들                |
|                  | client-dl      | 다른 엔티티들이 클라이언트로부터 다운로드 할 클래스 파일들 |
|                  | Policy         | 서비스될 클래스 파일들의 접근의 허용에 관련된 파일들    |
|                  | Service        | 서비스의 구현 클래스 파일들                  |
|                  | Service-dl     | 서비스가 다른 엔티티들에게 다운로드 가능한 클래스 파일들  |
|                  | Storage        | 서비스 요청 수신시 서비스와 관련된 클래스 파일 생성    |
|                  | Temp           | 임시 로그 저장 파일                      |

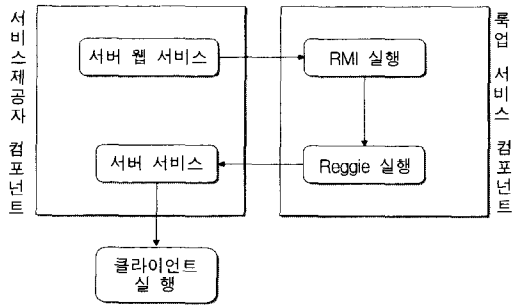
만약 서비스 제공자와 록업 서비스가 서비스를 위한 클래스 파일을 임의의 디렉토리로 구성하더라도 JSMS에서 제안하고 있는 지니 서비스 파일 시스템 구조에 맞게 자동으로 환경을 재구성하고 설정한다.

4. 지니 서비스 관리 시스템의 구현

4.1 서비스 컴포넌트

본 논문에서 제안하고 있는 JSMS는 (그림 7)에서 보는 바와 같이 서비스 제공자 컴포넌트와 록업 서비스 컴포넌트로 구성되어 있다. 서비스 제공자 컴포넌트는 웹 서비스와 서비스 제공자가 클라이언트에게 제공하는 서버 서비스

를 실행한다. 룩업 서비스 컴포넌트는 rmid 데몬 서비스와 reggie 서비스와 같은 서비스 관리자 역할을 실행한다. 이들 각 컴포넌트는 지니 시스템 서비스의 관리를 위한 자동화 모듈을 포함하고 있으며, 실행시에 구동되어 지니 시스템 서비스 환경을 구축한다.

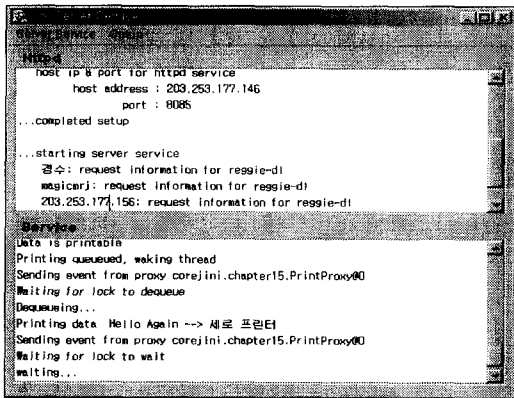


(그림 7) JSMS 기반 지니 시스템의 서비스 흐름도

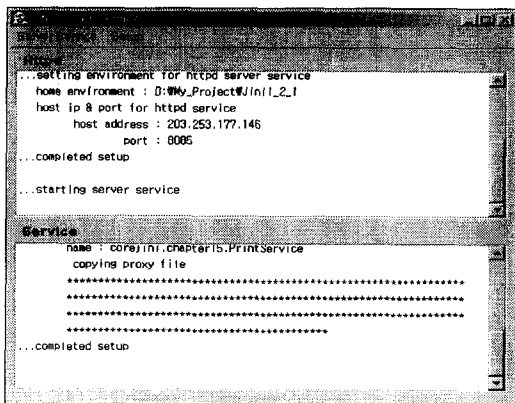
서비스는 룩업 서비스와 클라이언트가 서비스 제공자인 HTTP 서버에 접근할 수 있도록 하는 톱로 역할을 한다. 서버 서비스는 룩업 서비스와 클라이언트에게 서비스를 등록하고 실행 가능하도록 클래스 파일을 쉽게 다운로드 할 수 있게 한다. 실제 지니 네트워크에 제공되는 서비스 프록시 클래스 파일을 웹 서비스의 루트 디렉토리에 위치시킨다. 서버 서비스는 서비스 제공자가 지니 네트워크상에서 서비스를 사용할 수 있도록 룩업 서비스를 찾는다. 그러면 룩업 서비스는 웹 서비스의 루트 디렉토리에 위치한 서비스 프록시 클래스 파일을 다운로드하여 룩업 서비스의 서비스 리스트에 등록시킨다. 그리고 클라이언트에서 서비스에 대한 요청이 있을 때까지 대기한다.

(그림 8)은 서비스 제공자 컴포넌트를 실행한 것으로, 서버 시스템의 초기화 및 자동화 모듈 실행으로 환경이 자동 설정되는 부분 (그림 8)(a)과 클라이언트가 서비스 제공자에게 서비스 요청시 서버 서비스의 실행 부분 (그림 8)(b)이다. 만약 자동화 모듈이 없는 기존의 지니 시스템 환경에서의 서비스 실행을 위해서는 사용자나 관리자가 지니 라이브러리 위치정보, 네트워크 정보, 웹 서비스 정보, 서비스를 제공하기 위한 클래스 정보등을 직접 설정해야 한다.

4.2 서비스 제공자 컴포넌트



(a) 자동화 모듈 실행 및 서버 시스템 초기화

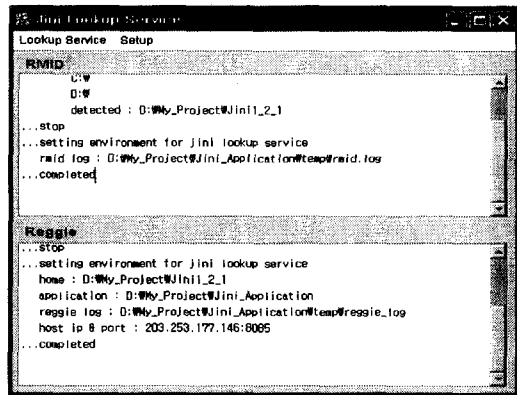


(b) 서버 서비스 실행

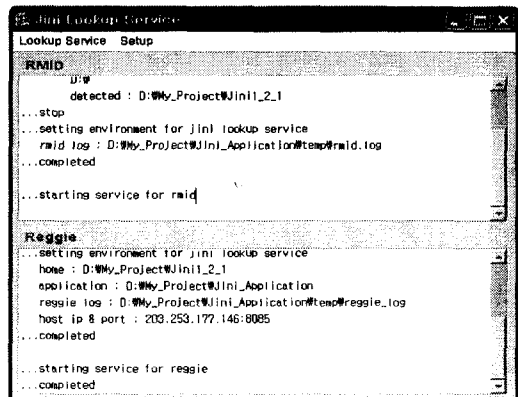
(그림 8) 서비스 제공자 컴포넌트 실행

서비스 제공자에서 실행되어야 할 서비스 제공자 컴포넌트는 서비스 프록시 클래스 파일의 다운로드가 가능하도록 하는 웹 서비스와 서비스를 제공하는 서버 서비스이다. 웹

4.3 룩업 서비스 컴포넌트



(a) 자동화 모듈 실행 및 룩업 서비스 시스템 초기화



(b) 룩업 서비스 실행

(그림 9) 룩업 서비스 컴포넌트 실행

록업 서비스를 수행하는 록업 서비스 컴포넌트는 rmid와 reggie 로 구성되어 있다. 지니 네트워크 상에서 서비스 제공자를 찾고, 서비스 제공자에서 제공하는 서비스 프록시 클래스 파일을 다운로드하여 록업 서비스의 서비스 리스트에 등록시킨다. 지니 네트워크에서는 서비스에 대한 요청과 처리 결과를 RMI 통신을 사용해서 처리하고, RMI 통신을 원활하게 할 수 있도록 해주는 것이 RMI 데몬이다.

Reggie는 서비스 제공자로부터 제공받은 서비스 프록시 클래스 파일을 획득하여 서비스 리스트에 등록시키고, 여러 개의 서비스들을 관리하는 역할을 한다. 이 Reggie를 실행하기 위해서는 반드시 서비스 제공자의 웹 서비스로부터 'reggie.jar'와 'reggie-dl.jar'에 대한 정보를 받아와야 한다.

(그림 9)은 록업 서비스 컴포넌트를 실행한 것으로, 록업 서비스 시스템의 초기화 및 자동화 모듈 실행으로 rmid와 reggie의 실행환경이 자동 설정되는 부분 (그림 9)(a)과 rmid 서비스와 reggie 서비스의 실행 부분(그림 9)(b)이다.

#### 4.4 실험 성능 및 평가

JSMS는 Windows 2000 운영체제, JDK1.3.1과 Jbuilder7 자바 개발툴, 그리고 JINI 1.2.1 라이브러리를 사용한다. 테스트 환경은 네트워크로 연결되어 있는 3대의 컴퓨터를 사용하고 있으며 windows 2000과 windows XP 운영체제가 설치되어 있다. 그리고 컴퓨터들은 서비스 제공자, 록업 서비스 그리고 서비스 이용자로서 실행한다. 테스트는 4.1절 (그림 7)의 지니 시스템의 서비스 흐름에 따라 실행한다. <표 3>은 지니 시스템의 기본적인 실행환경과 본 논문에서 제안된 JSMS 실행환경에서의 성능비교이다. <표 3>에서 보는 바와 같이, 로컬 컴퓨터의 지니 라이브러리와 파일 시스템 구조의 배치 및 실행방법 등은 관리자나 사용자가 직접 개입하여 실행환경을 설정해야 한다. 하지만 JSMS는 이러한 일련의 작업을 자동으로 설정한다. 이것은 로컬 컴퓨터의 정보를 획득하여 지니 서비스 환경을 맞추기 때문이다.

<표 3> 지니 시스템 실행환경 성능 비교

| 자동화 여부     | 지니 시스템의 기본적인 실행 환경 | 본 논문에서 제안된 JSMS 실행환경 |
|------------|--------------------|----------------------|
| 지니 라이브러리   | 수동                 | 자동                   |
| 지니 네트워크 정보 | 수동                 | 자동                   |
| 파일 시스템 구조  | 수동                 | 자동                   |
| 실행방법       | 수동                 | 자동                   |

현재 지니 시스템을 이용한 서비스를 제공하기 위해서는 지니 홈네트워크에 참여하는 구성요소의 시스템에 대한 지식과 정보를 사전에 필요로 한다. 즉, 서비스 제공자, 록업 서비스 그리고 클라이언트는 지니 시스템에 대한 컴포넌트와 서비스 아키텍처에 대한 사전지식을 요구하고, 이러한 사전지식이 있어야 지니 시스템 서비스의 실행이 가능하기

때문이다.

하지만 본 논문에서 제안한 JSMS는 지니 시스템 구성요소들 간의 상호 유동적인 서비스를 위해 기본적으로 사용되는 지니 서비스 정보 및 시스템 환경정보의 획득이 가능하다. 그래서 서버 서비스 컴포넌트와 록업 서비스 컴포넌트를 실행하는 서비스 제공자나 서비스 관리자는 지니 시스템에서 기본적으로 필요로 하는 지니 시스템 관련 정보나 클라이언트에게 제공할 서버 서비스 클래스 파일의 위치정보, 그리고 현재 실행하는 시스템의 네트워크 정보와 서비스 파일 시스템 구조 등과 같은 자세한 정보를 사전에 필요로 하지 않는다. 이는 실행시에 자동화 모듈을 통해서 정보 획득이 가능하고, 자동적으로 이러한 정보를 바탕으로 지니 시스템에 필요한 각종 환경에 설정된다. 따라서 서비스 관리자나 서비스 제공자는 단지 록업 서비스 컴포넌트와 서버 서비스 컴포넌트를 실행하여 사용자가 원하는 서비스를 선택하여 실행하면 되는 것이다.

#### 5. 결론 및 활용방안

앞으로 홈네트워크는 다양하고 복잡한 정보가전기기 및 네트워크 미디어와 프로토콜이 혼재하는 분산환경이 될 것이고, 이들 하드웨어적인 요소를 기반으로 서로 유연하게 상호작용하여 서비스를 이용하기 위한 보다 발전된 미들웨어 기술을 요구하게 될 것이다.

본 논문에서 제안한 JSMS(Jini Service Management System)는 기존의 지니 시스템에서 서비스 제공자나 록업 서비스의 관리자가 해야 하는 보다 복잡한 작업을 상당히 줄여줌으로써, 서비스 이용자나 서비스 제공자가 직접 개입해야 하는 부분을 최소화할 수 있다. 그리고 서비스 이용자에게 서비스를 이용할 수 있는 환경을 제공하기 위해 서비스 제공자나 록업 서비스의 관리자가 숙지해야 하는 어렵고 복잡한 서비스 실행 환경을 보다 단순화시킴으로써, 홈네트워크의 복잡성과 이질성으로 발생할 수 있는 복잡한 관리 기술의 요구에 유연한 미들웨어 기술로 발전시키는 자료로서 활용이 가능하다.

#### 참 고 문 헌

- [1] Edwards, W. Keith/ Edwards, W., Core Jini, Prentice Hall Computer, 2001.
- [2] Sing Li, Professional Jini, Wrox, 2000.
- [3] GERARD O. RISCOLL, The Essential Guide to Home Networking Technologies, Prentice Hall Computer Books, 2000.
- [4] "http://www.sun.com/software/jini/specs/index.html," Jini Specification, Jini Home Page.
- [5] "http://java.sun.com/products/jdk/rmi," Java RMI Specification and White paper, Java Home Page.

[6] "http://www.artima.com/index.jsp." a resource for Java and Jini developers.

[7] "http://pandonia.canberra.edu.au/java/jini/tutorial.1.03/Jini.xml," Jan Newmarch's Guide to JINI Technologies, Dec., 1999.

[8] "http://www.cswl.com/whitepr/tutorials/jini.html," Jini by Example White paper, May., 1999.

[9] "http://www.upnp.org/resources/whitepapers.asp," UPnP White paper, UPnP Forum.

[10] Lisa A. Schaefer, "ARCHITECTURE USING JINI TECHNOLOGY FOR SIMULATION OF AN AGENT-BASED TRANSPORTATION SYSTEM," Proceedings of the 33rd conference on Winter simulation, IEEE Computer Society, pp.1079-1083, 2002.

[11] Matt Storey, Gordon Blair, Adrian Friday, "MARE : Resource Discovery and Configuration in Ad Hoc Networks," Mobile Networks and Applications, Vol.7, No.5, pp.377-387, 2002.

[12] Pascal Ledru, "Smart Proxies for Jini Services," ACM SIGPLAN Notices, Vol.37, No.4, pp.57-61, 2002.

[13] Christopher Dabrowski, Kevin Mills, Jesse Elder, "Understanding Consistency Maintenance in Service Discovery Architectures during Communication Failure," Proceedings of the third international workshop on Software and performance, pp.168-178, 2002.

[14] C. Dabrowski and K. Mills, "Understanding Self-healing in Service-Discovery Systems," Proceedings of the first workshop on Self-healing systems, pp.15-20, 2002.

[15] Tatsuo Nakajima, "A Middleware Component Supporting Flexible User Interaction for Networked Home Appliances," ACM SIGARCH Computer Architecture News, Vol.29, No.5, pp.68-75, 2001.

[16] Adrian Friday, Nigel Davies, Elaine Catterall, "Supporting Service Discovery, Querying and Interaction in Ubiquitous Computing Environments," Second ACM international workshop on Data engineering for wireless and mobile access, pp.7-13, 2001.

[17] Nigel Davies, Pierre-Guillaume Raverdy, "POSITION PAPER : THE ROLE OF PLATFORMS AND OPERATING SYSTEMS IN SUPPORTING HOME NETWORKS," Proceedings of the 9th workshop on ACM SIGOPS European workshop, pp.165-170, 2000.

[18] Jim Waldo, "THE JINI ARCHITECTURE FOR NETWORK-CENTRIC COMPUTING," COMMUNICATIONS OF THE ACM, Vol.42, No.7, pp.76-82, ACM Press, July, 1999.

[19] Ken Arnold, "The Jini Architecture : Dynamic Services in a Flexible Network," Proceedings of the 36th ACM/ IEEE conference on Design automation conference, pp.157-162, 1999.

[20] 문경덕, 배유식, 김채규, "홈네트워크 제어 미들웨어 개요 및 표준화 동향", 정보처리학회지, 제8권 제5호, pp.45-52, 2001.



**정 준 영**

e-mail : mrj@hawk.com.kyungnam.ac.kr  
 1999년 경남대학교 전산통계학과(학사)  
 2001년 경남대학교 대학원 컴퓨터공학과  
 (공학석사)  
 2001년~현재 경남대학교 컴퓨터공학과  
 박사과정

관심분야 : 자바기술, 홈네트워킹, 임베디드시스템 등



**정 민 수**

e-mail : msjung@eros.kyungnam.ac.kr  
 1986년 서울대학교 컴퓨터공학과(학사)  
 1988년 한국과학기술원 전산학과(공학석사)  
 1994년 한국과학기술원 전산학과(공학박사)  
 1988년~1990년 한국과학기술원 전산학과  
 T.A. 및 R.A

1990년~현재 경남대학교 정보통신공학부 교수  
 관심분야 : 자바기술, 홈네트워킹, 임베디드시스템 등



**김 광 수**

e-mail : kskim@keri.re.kr  
 1983년 서울대학교 전기공학과(학사)  
 1985년 서울대학교 대학원 전기공학과  
 (공학석사)  
 1998년 Texas A&M Univ., Electrical  
 Eng. Ph.D.

1986년~현재 한국전기연구원 책임연구원  
 관심분야 : 임베디드시스템, 자바기술, 광통신 등