

논문 2005-42TC-2-1

모바일 환경을 위한 Jini-Agent 시스템 구현

(Implementation of Jini-Agent System for Mobile Environments)

김 상 태*, 윤 병 주**, 박 원 배**, 김 현 덕*

(Sang-Tae Kim, Byoung-Ju Yun, Won-Bae Park, and Hyun-Deok Kim)

요 약

본 논문에서는 리소스가 제한되는 모바일 환경에서 Jini 서비스를 사용할 수 있도록 Jini-Agent 시스템을 구현하였다. 구현된 Jini-Agent 시스템은 모바일 디바이스에 별도의 클라이언트 프로그램을 설치할 필요 없이 Jini 접속 기술을 지원 할 수 있도록 한다. 또 Jini-Agent 시스템에 dynamic service list 기능을 구현하여 웹 브라우저를 통해 실시간으로 편리하게 서비스를 검색하고 이용할 수 있도록 하였다. Jini-Agent를 이용한 응용 모델로서 모바일 디바이스를 위한 파일 검색 서비스와 프린트 서비스를 구현하였다.

Abstract

We have implemented a Jini-Agent system which enables resource-limited mobile devices to utilize the Jini connection service. The Jini-Agent system supports Jini connection technologies to mobile devices without additional client program installation in it. It also supports a dynamic service list which helps the mobile devices to search and to utilize the services through web browser in real time. A couple of examples such as a file search service and a print service with mobile devices have been demonstrated by using the Jini-Agent system.

Keywords : Jini-Agent system, Dynamic service list, Mobile device

I. 서 론

최근 홈네트워크 시장이 급속하게 성장하고 있다. 홈네트워크 서비스는 사용자가 필요한 서비스를 쉽게 검색하고, 네트워크에 대한 사전 지식 없이도 원하는 서비스를 쉽게 이용할 수 있도록 규격화된 미들웨어를 필요로 한다. 이러한 요구조건을 충족하기 위해 Sun Microsystems에서 차세대 홈네트워크용 미들웨어로 Jini 접속 기술^[1]을 제안하였다.

Jini 접속 기술은 네트워크에 연결된 주변기기나 응

용프로그램들을 사용자가 쉽게 발견하고 이용할 수 있는 네트워크 플러그 앤 워크(Network Plug and Work)를 제공한다. 현재 Jini 접속 기술의 이러한 장점을 이용하여 홈네트워크에 필요한 미들웨어 기술들을 데스크탑 환경뿐만 아니라 모바일 환경에서도 동일한 서비스를 이용하고자 하는 사용자들의 요구가 증가하고 있다. 즉, 모바일 디바이스에서 Jini 접속 기술을 수용할 경우, 각종 Jini 서비스들을 쉽게 이용할 수 있다. 그러나 PDA나 휴대폰과 같은 모바일 디바이스들은 각각 고유의 특성과 휴대의 편리함 등을 이유로 최소한의 리소스만 제공한다. 따라서 데스크탑 환경을 기반으로 하여 설계된 Jini 접속 기술을 지원하기에는 많은 제약이 따른다.

한편, 리소스가 제한되는 모바일 환경에 적합하도록 모바일 디바이스용 플랫폼으로 J2ME(Java2 Platform, Micro Edition)^[2]가 제안되었다. 하지만, J2ME는 RMI (Remote Method Invocation)^[3] Profile을 지원하지 않음

* 정회원, 경북대학교 전자공학과
(Department of Electronic Engineering, Kyungpook National University)

** 정회원, 경북대학교 정보통신학과
(Department of Information and Communication, Kyungpook National University)

※ 이 논문은 2003년도 경북대학교의 연구비에 의하여 연구되었음

접수일자: 2004년11월12일, 수정완료일: 2004년12월23일

므로 RMI 통신을 기반으로 하는 Jini 접속 기술을 지원할 수 없다. 이러한 J2ME의 제약 때문에 J2ME 플랫폼을 적용한 모바일 디바이스에서는 기존의 Java 분산 네트워크 기술을 적용한 Jini 서비스들을 사용할 수 없는 문제점이 있다.

본 논문에서는 이러한 점을 고려하여 리소스 제한으로 인해 J2ME 기술을 수용하고 있는 모바일 디바이스가 Jini 서비스를 이용할 수 있도록 Jini-Agent 시스템을 구현하였다. 또, 웹 브라우저를 통해 Jini 서비스들을 쉽게 이용할 수 있도록 Dynamic service list 기능을 제공하도록 하였으며, 응용 모델로서 모바일 디바이스를 위한 파일 검색 서비스와 프린트 서비스를 구현하였다.

본 논문의 구성은 다음과 같다. II장에서는 차세대 미들웨어 기술인 Jini 기술과 Sun사에서 나온 소형 디바이스들을 위한 플랫폼인 J2ME에 대해서 알아본다. III장에서는 구현한 시스템의 구성과 동작과정에 대해서 설명하고, IV장에서 결론을 맺는다.

II. 본 론

1. Jini 접속 기술

Jini 접속 기술은 Java를 기반으로 설계되어 있으며, 객체지향 분산 시스템을 구축할 수 있는 미들웨어이다. Jini 접속 기술을 이용하면 네트워크에 연결된 디바이스나 애플리케이션들은 하나의 서비스 개념으로 처리되므로, 네트워크를 통해 특별한 구성이나 드라이버의 설치 없이 이용할 수 있다. 또한 Jini 네트워크는 중앙 집중식의 컨트롤이 아니므로, 디바이스가 Jini 네트워크에

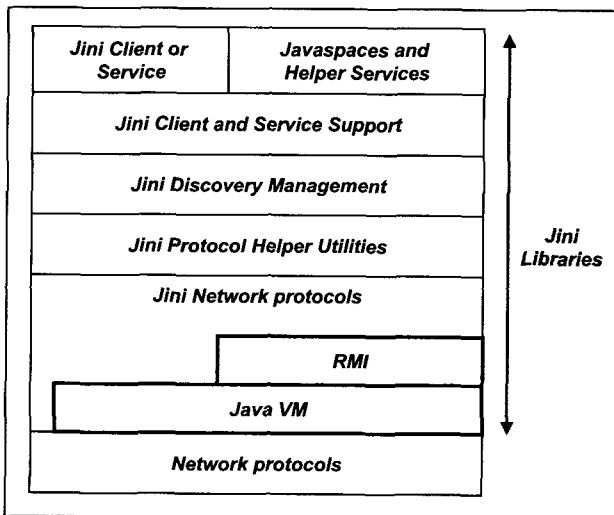


그림 1. Jini 네트워크 하부 구조
Fig. 1. Layer of Jini network.

참여했을 때 전체 네트워크는 동적으로 관리된다. Jini 네트워크의 하부 구조는 그림 1과 같이 네트워크 프로토콜을 기반으로, non-Java 디바이스들과 통신할 수 있는 Jini 네트워크 프로토콜을 제공하고 있다. 그리고 사용 가능한 서비스들의 API(Application Programming Interface)를 제공하며 Discovery, Lookup과 같은 Helper 서비스를 제공한다^[4].

Jini 시스템을 구성하는 기본 요소는 그림 2와 같다. Jini 네트워크에 연결된 서비스 제공자(Service Provider)와 서비스를 이용하는 서비스 이용자(Service Client), 그리고 Jini 시스템을 전체적으로 관리, 운영 및 제어하는 서비스 관리자(Lookup Service)로 구성 되어 있다^[5].

각각 구성요소들의 동작 과정은 그림 3과 같이 서비스 제공자가 Lookup Service를 찾는 Discovery 과정,

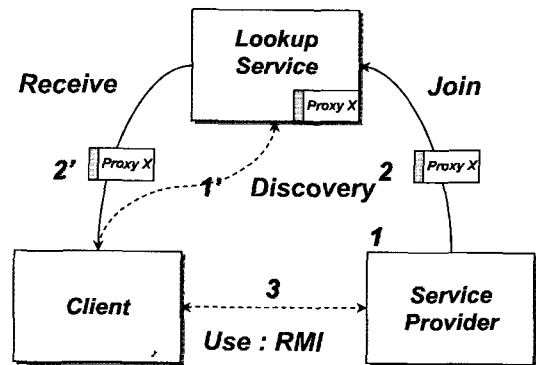


그림 2. Jini 네트워크의 상호작용
Fig. 2. Components of Jini network.

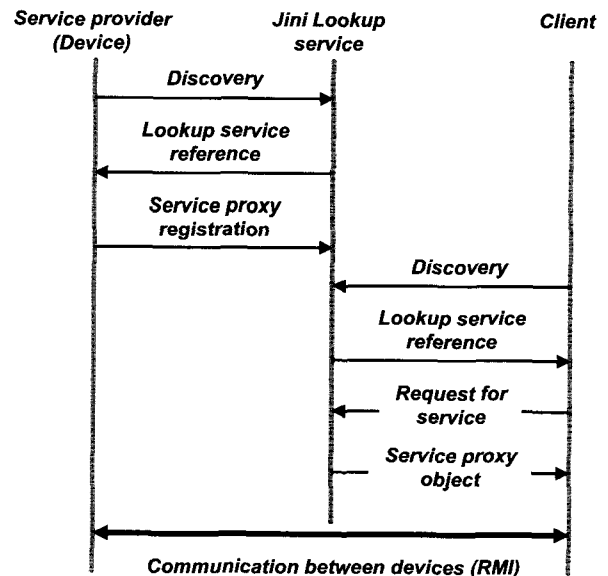


그림 3. Jini 네트워크 동작 다이어그램
Fig. 3. Processing of Jini network.

서비스 자신을 등록하는 Join 과정, 그리고 Lookup Protocol을 사용하여 서비스 이용자와 서비스 제공자가 상호 통신이 이루어지는 부분으로 구성된다.

서비스 이용자는 클라이언트와 Back-End 서비스간의 통신 역할을 하는 Java Object(proxy)^[6]를 통하여 서비스 제공자와 Remote 서비스 호출을 위해 RMI 통신을 한다.

또한, Jini 접속 기술은 Java를 기반으로 했기 때문에 플랫폼이 독립적이어서 이식성이 우수하고, RMI 전송 방법을 채택하여 보안성이 우수하다. 그러나 제한된 리소스를 가진 모바일 디바이스에서 Jini 접속 기술을 수용하기 위해서는 추가적인 자원이 필요하다. 대부분의 모바일 디바이스는 Jini 접속 기술을 수용하기 위한 자원이 부족한 상황이며, 비교적 충분한 자원을 가지고 있더라도 데스크 탑 환경을 중심으로 설계된 Jini 접속 기술을 수용하기에는 많은 문제가 있다. 따라서 모바일 디바이스에서 Jini 접속 기술을 위한 자원의 추가를 최소화 하면서 Jini 서비스를 쉽고 편리하게 이용할 수 있는 Jini 네트워크 연동 기술의 개발이 필요하다.

2. Java Micro Edition

Java의 표준 플랫폼은 일반 PC 환경에서 접하는 J2SE(Java2 Standard Edition)와 서버 환경을 위한 J2EE(Java2 Enterprise Edition), 그리고 모바일 환경을 위한 J2ME(Java2 Micro Edition)로 구분된다.

J2ME는 일반 PC나 서버 컴퓨터와 같은 환경이 아닌 제약된 환경, 즉 모바일 기기나 가전 등에서 Java 응용 프로그램을 실행하기 위해 개발된 소형 플랫폼이다.

일반 Java 플랫폼에서는 Java 프로그램을 실행시키기 위해 VM(Virtual Machine)으로 JVM(Java VM)을 이용하지만, J2ME에서는 그림 4와 같이 기존 JVM이 아닌 모바일 디바이스들을 위한 두 개의 JVM을 이용한다.

그 중 하나는 KVM(Kilobyte VM)이고 다른 하나는 CVM(Classic VM)이다.

KVM은 16~32bit의 프로세스와 160~512K의 메모리를 가진 디바이스를 위한 VM인 반면, CVM은 32bit 이상의 프로세스와 2MB이상의 메모리를 가진 디바이스에서 사용된다^[7]. J2ME 프로그래밍이 적용되는 디바이스들은 다시 디바이스들의 특징에 따라 최소한의 사양만을 정의 해 놓은 Configuration과 그 위에 다양한 디바이스 환경에 맞도록 구체화 시킨 Profiles로 구분된다.

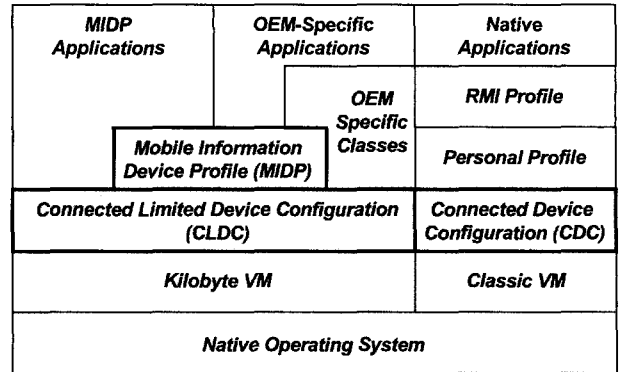


그림 4. J2ME 구조

Fig. 4. Mobile information device profile architecture.

표 1. CDC와 CLDC

Table 1. Comparison of CDC and CLDC.

	CLDC(Connected Limited Device Configuration)	CDC(Connected Device Configuration)
CPU	16~32bit	32bit 이상
Memory	160~512K	2M 이상
VM	KVM	CVM
Device	핸드폰, 가전제품, PDA	셋톱박스 TV, 자동차 항법시스템

J2ME에서 정의하는 Configuration에는 표 1과 같이 CLDC(Connected Limited Device Configuration)와 CDC(Connected Device Configuration) 두 가지로 나눌 수 있다.

CLDC는 개인 휴대가 가능하며 네트워크를 통한 정보 전달이 가능한 디바이스들로, 휴대폰이나 PDA 등이 여기에 속하고, CDC는 CLDC보다 덜 제한적인 환경을 가진 디바이스들로, 셋톱박스나 인터넷 TV, 자동차 네비게이션 시스템 등이 이에 포함된다.

모바일 환경의 디바이스들은 각각의 고유 특성과 휴대의 편리함 등을 이유로 최소의 자원을 가진다. 그래서 Jini에서 지원되는 부동 소수점, Reflection, 에러 처리 기능들을 사용하는데 여러 가지 문제가 발생하게 된다^[8]. 이러한 문제들을 좀 더 자세히 살펴보면 다음과 같다.

가. 부동 소수점을 지원하지 않는다.

일반 데스크 탑 환경과 달리 모바일 디바이스에서는 부동 소수점을 지원하지 않는다. 이는 모바일 디바이스가 하드웨어적으로 부동 소수점을 지원하지 않기 때문

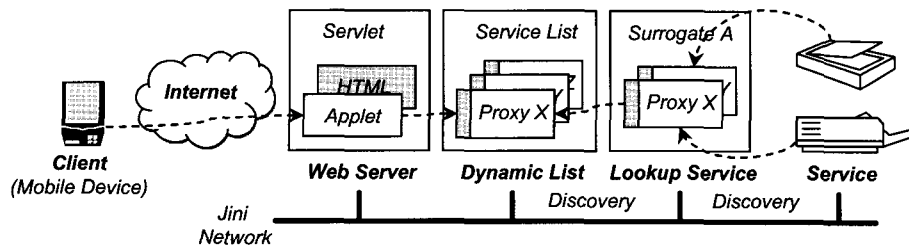


그림 5. Jini-Agent 시스템 구성도
Fig. 5. Jini-Agent system architecture.

에 소프트웨어적으로 부동 소수점을 지원하는 것은 너무 큰 부하가 있다. 그래서 Virtual Machine에서 부동 소수점을 지원하지 않는다.

나. Reflection을 지원하지 않는다.

Reflection은 Runtime시 Virtual Machine 내부의 클래스에 대한 정보들을 가지고 있다. 하지만 모바일 디바이스를 위한 Virtual Machine에서는 Reflection을 사용할 수 없다. 이는 Reflection 기능을 이용하는 Java의 RMI, Jini 등은 모바일 환경에서는 사용할 수 없음을 의미한다.

다. 제한적인 Error 처리 지원

Java에서는 Error와 Exception 두 가지의 예외 상황을 처리한다. 모바일 디바이스를 위한 Virtual Machine에서는 Exception 처리는 지원하지만 Error 처리는 제한적으로 지원한다. KVM에서는 세 가지 Error 처리만을 수행하는데, 이렇게 대부분의 Error 처리가 빠져 버린 이유 중의 하나는 Error 처리가 일관성이 없는 모바일 환경 하에 있는 대부분의 디바이스들은 Error가 발생했을 때 단순히 리셋만으로 Error 처리가 매우 어렵고, 너무 심한 오버헤드를 초래할 가능성이 높기 때문이다.

라. JNI(Java Native Interface)를 지원하지 않는다.

모바일 디바이스를 위한 Virtual Machine에서는 JNI를 지원하지 않는다. JNI는 용량이 크기도 하지만, KVM의 제한적인 보안 모델로 인해, Native 함수의 호출 자체가 위험할 수 있기 때문에 지원하지 않는다.

이러한 J2ME의 제약사항에 의해서 기존의 Jini와 같은 Java의 분산 네트워크 미들웨어 기술 적용에 필요한 자원이 부족한 상황이며, 최근에 개발되는 모바일 디바이스들은 비교적 충분한 자원을 가지고 있지만, 모바일 환경에서의 Jini 접속 기술에 대한 연구가 부족한 상황

이다.

III. 시스템 구현

Jini 네트워크는 데스크 탑 환경을 대상으로 하기 때문에 제한된 자원을 가진 모바일 디바이스들을 수용하지 못한다. 이 문제를 해결하기 위해 본 논문에서는 모바일 환경의 디바이스들이 Jini 서비스를 이용할 수 있도록 모바일 디바이스를 위한 Jini-Agent 시스템을 구현하고, 그 응용 모델로써 파일 검색 서비스와 프린트 서비스를 구현하였다.

1. 구성

본 논문에서 제안된 시스템의 구성도는 그림 5와 같이 크게 네 부분으로 구성된다.

Jini 네트워크에 서비스를 제공하거나 이용하려는 디바이스들이 Proxy를 통해 Lookup Service에 등록하는 부분, Lookup Service에 등록되어 있는 Jini 서비스들을 검색하여 보여 주는 Dynamic service list 부분, Dynamic service list에서 선택한 서비스의 Proxy를 복사하여 Servlet으로 전달하는 부분, 그리고 Servlet을 통해 Jini 네트워크에 등록되어 있는 서비스를 이용하는 모바일 디바이스 부분으로 이루어진다.

2. 동작원리

시스템의 내부 동작 원리는 그림 6과 같다.

먼저 서비스 제공자는 Discovery Protocol을 이용하여 가까운 Lookup 서비스에 Registrar를 이용하여 Proxy 객체의 복사본을 등록한다. 이때 Jini가 지원되지 않는 디바이스가 Jini 네트워크에 연결되면 Surrogate가 대신하여 Lookup 서비스에 등록한다. Lookup 서비스에 등록된 Jini 서비스들의 목록들은 Dynamic service list를 통해 웹 서버의 Servlet으로 전달된다.

서비스 이용자는 Jini 서비스를 이용하기 위해 웹 서

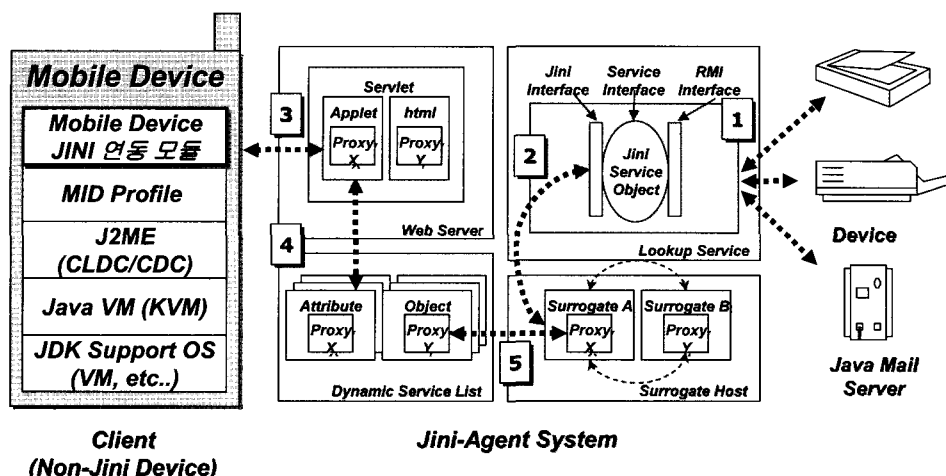


그림 6. Jini-Agent 시스템 동작 과정
Fig. 6. Internal structure of Jini-Agent system.

버의 Servlet을 동작 시키고, Servlet은 Applet/html을 통하여 Jini 서비스를 제공한다. 그리고 Servlet에 의해 얻어진 서비스들에 대한 결과 값은 Dynamic service list를 통하여 웹 브라우저가 있는 서비스 이용자의 모바일 디바이스 상에 보여 지게 된다.

이 과정에서 웹 브라우저가 있는 서비스 이용자는 Servlet이 동작하는 사이트의 URL을 통하여 원하는 Jini 서비스를 이용할 수 있으며, 사용자들은 별도의 클라이언트 프로그램을 설치할 필요 없이 편리하게 사용할 수 있다.

3. 구현환경

본 논문에서 구현된 시스템 환경은 다음과 같다.

- OS: Windows 2000 Server
- Java: J2sdk 1.3.1, Personal Java
- Jini: Jini 1.1
- Servlet: Jsdk 2.1
- Jeode VM^[9]
- Mobile Device: Compaq PDA(pocket pc H3600)

IV. 시스템 구현 결과

구현된 Jini 서비스를 모바일 디바이스에서 언제, 어디서나 이용할 수 있도록 하기 위하여 웹 서버로 Servlet을 이용한다. Servlet에서 제공하는 Applet/html을 이용하기 위하여 웹 브라우저가 탑재된 WinCE 계열의 Pocket PC PDA를 이용한다.

Jini-Agent를 이용한 응용 모델의 시나리오는 다음과

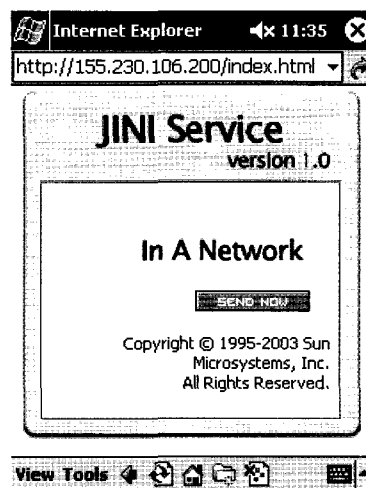


그림 7. 모바일 Jini-Agent 시스템 Main 화면
Fig. 7. The initial screen of Jini-Agent12.

같다.

먼저 모바일 디바이스를 사용하여 Jini 네트워크에 등록되어 있는 Host에 접속한 후, 사용자가 원하는 특정 문서를 찾는다. 문서를 찾아 읽은 후, Jini 서비스에 등록된 프린터로 인쇄를 한다.

실제로 모바일 디바이스를 사용하여 Jini 네트워크에 등록된 서비스를 이용하기 위하여 실행시킨 화면은 다음과 같다.

① 그림 7은 실제 PDA 상의 웹 브라우저를 이용하여 Servlet이 동작하는 웹 서버의 URL을 통하여 Jini 네트워크에 접속한 화면이다.

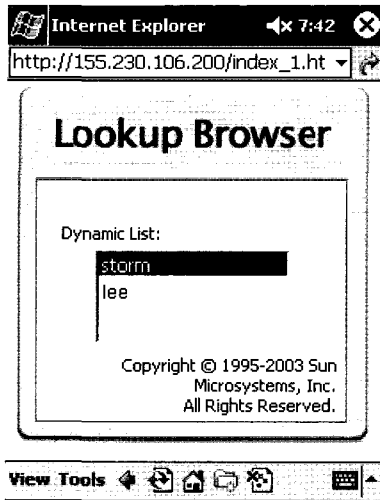


그림 8. Dynamic service list 화면
Fig. 8. The screen of Jini service by Dynamic service list.

② 그림 8은 Lookup 서버에 등록 되어 있는 Jini 서비스들의 Proxy 값을 다운로드하여 Jini 서비스 목록들을 동적으로 보여주는 Dynamic service list 화면이다.

Dynamic service list 화면을 제공함으로써 클라이언트는 Jini 네트워크에 등록 되어 있는 현재 사용 가능한 Jini 서비스들을 편리하게 검색할 수 있다.

Dynamic service list의 Source Code는 다음과 같다.

```
import java.util.ArrayList;
public class Client implements Runnable
{
    // 서비스 List를 저장하기 위한 Vector 선언
    private java.util.List vec = new ArrayList();
    class Discoverer implements DiscoveryListener
    {
        // Lookup Service에 등록된 서비스들의 List를 저장
        vec.add(newregs[i].getLocator().getHost()+" ");
        for(int j = 0; j < vec.size(); j++) {
            dic += (String)vec.get(j);
        }
        System.out.println("dic:newregs["+i+"].getLocator().getHost() = " + dic);
    }
}
```

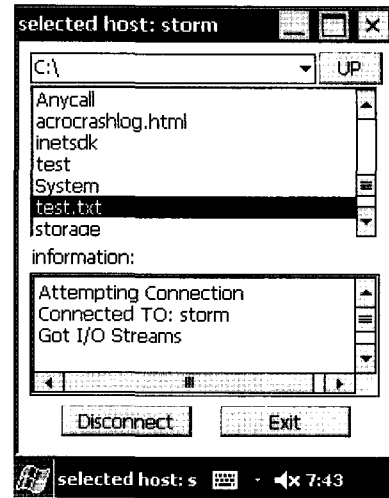


그림 9. 특정 서비스 파일 검색 화면
Fig. 9. The screen of connection to a particular service.

③ Jini 네트워크에 등록 되어 있는 특정 서비스로 접속을 허용하기 위하여, 모바일 디바이스 내의 Java Security 파일을 생성하는 과정이 필요하다.

먼저 자원 접근 허용을 제공하기 위한 mypolicy 정책파일을 생성하는 코드이다.

```
grant {
    permission java.security.AllPermission;
};
```

그리고 생성된 정책파일을 모바일 디바이스에 상의 java.security 내에 등록을 시켜준다.

```
policy.url.3=file:/Windows/lib/security/mypolicy
```

이렇게 보안파일을 생성하고 등록함으로써, 모바일 디바이스에서 Jini 네트워크에 등록된 특정 서비스로 접근이 가능해진다.

④ 그림 9는 Dynamic service list에 등록된 Jini 서비스 중에서 특정 서비스를 선택하여 접속한 화면이다. 접속이 이루어지면 서비스내의 디렉토리 와 파일 정보를 찾아 보여준다.

⑤ 그림 10은 서비스 내의 파일 중에서 특정 문서 파일 선택하여 읽어 들인 화면이다.

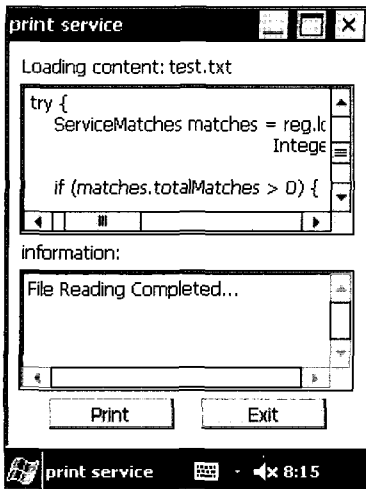


그림 10. 특정 파일을 읽어 들인 화면

Fig. 10. The screen of reading file.

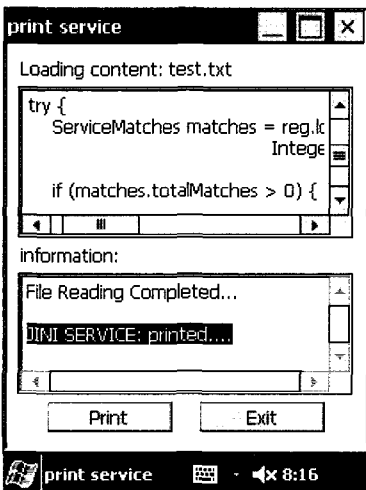


그림 11. 특정 파일을 읽은 후 Print 수행 화면

Fig. 11. The screen of print operation.

문서를 읽은 후, 인쇄 버튼을 누르게 되면 Lookup Service에 등록되어 있는 프린터로 데이터를 전달하게 된다.

⑥ 데이터가 성공적으로 프린터로 전달되고, 프린터에서 인쇄가 수행되면, 프린터 서버에서는 응답메시지를 Jini Lookup 서버로 전달하게 된다.

Servlet은 Jini Lookup 서비스에 등록 되어 있는 프린터 서비스의 Proxy와 서비스 ID를 Jini-Agent로부터 받고, 사용자가 원하는 서비스와 일치하는 프린터 서비스를 찾는다. 그리고 일치되는 프린터 서비스에 대한 속성 값을 얻고, 특정 문서 파일의 내용을 프린터 서버



그림 12. Servlet 동작 화면

Fig. 12. The screen of Servlet.

로 전달한다. 마지막으로 프린터를 동작시켜 선택한 파일의 내용을 인쇄를 한다. 그림 11은 응답 메시지를 받은 화면이다.

그림 12는 Lookup Service에 등록 되어 있는 프린터 서비스를 찾고 수행하는 Servlet의 동작 화면이다.

V. 결 론

휴대폰, PDA 등의 모바일 디바이스들은 리소스가 제한되므로 J2ME 플랫폼에 기반 하여 구현되므로 홈네트워킹용 미들웨어인 Jini 서비스를 이용하기에는 많은 제약이 따른다. 즉, 대부분의 모바일 디바이스들은 Jini 접속기술 수용에 필요한 자원이 부족한 상황이며, 최근에 개발되는 디바이스들은 비교적 충분한 자원을 가지고 있지만 모바일 환경에서의 Jini 접속기술에 대한 연구가 부족한 상황이다.

본 논문에서 이러한 문제점을 해결하기 위해 Jini-Agent 시스템을 구현하였다. 구현된 시스템은 Jini 서비스를 지원 하지 않는 J2ME 플랫폼과 Jini 네트워크와의 연동을 위해 웹 서버에 Servlet을 두어 모바일 디바이스에서 Applet/html을 통하여 Jini-Agent에 접속하는 기술을 구현하였다.

Jini-Agent 시스템은 non-Jini 디바이스를 위한 Jini 네트워크 접속 기술이며, Lookup 서버에 등록된 서비스들을 Servlet으로 전달하여 모바일 디바이스에 제공함으로써 홈 네트워크 기기들과의 통신을 가능하게 한다. 또한 Jini 네트워크에서 제공하는 분산 서비스를 모바일 디바이스에서도 이용이 가능하게 해 준다.

이 시스템은 향후 Jini 서비스를 무선 환경에서 가정 내의 다양한 가전기기뿐만 아니라, 다른 미들웨어들과

의 연동 등을 통한 실생활에 적용될 수 있는 시스템을 보다 쉽게 만드는데 기반이 될 수 있을 것이다.

참고 문헌

- [1] Sun Microsystems, "Jini Specifications v. 1. 2," <http://www.sun.com/software/jini/specs/>, 2001.
- [2] Sun Microsystems, "Java 2 Platform Micro Edition," <http://java.sun.com/j2me/index.jsp>.
- [3] Sun Microsystems, "Java Remote Method Invocation Specification," <http://www.javasoft.com/product/jdk/rmi/>, Dec, 1999.
- [4] Sing Li, "Professional Jini," pp. 17-18, Feb, 2000.
- [5] Scott Oaks, Henry Wong, "Jini in a Nutshell," O'Reilly Press, Feb, 2000.
- [6] Sun Microsystems, "Java Object Serialization," Feb, 1998.
- [7] John W. Muchow, "Core J2ME™," Prentice Hall PTR, Jan, 2002.
- [8] Hinkmond Wong, "Developing Jini™ Applications Using J2ME Technology," Pearson Education, March, 2002.
- [9] Jeode VM, "Jeode™ Platform White Paper," Mar, 2001.

저자 소개



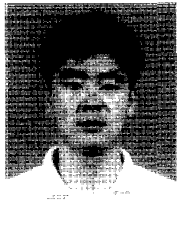
김상태(정회원)
2000년 대구대학교
정보통신학과 학사
2004년 경북대학교
정보통신학과 석사
2004년~현재 경북대학교
전자공학과 박사과정
<주관심분야 : 통신시스템, 광통신, 광네트워크>



윤병주(정회원)
1993년 경북대학교
전자공학과 학사
1996년 한국과학기술원
전기 및 전자공학과 석사
2002년 한국과학기술원
전자전산학과 박사
1996년~2003년 하이닉스 반도체 시스템
IC사업부 선임 연구원
2003년~현재 경북대학교 차세대정보통신연구소/
정보통신학과 조교수
<주관심분야 : 통신, 컴퓨터, 신호처리, 반도체>



박원배(정회원)
1989년 경북대학교
전자공학과 석사
1991년 Northrop University
전자공학 석사
1996년 Georgia 공대
Electrical & Computer
Engineering 박사
1997년~1999년 삼성전자 정보통신총괄 네트워크
사업부 선임 연구원
1999년~현재 경북대학교 차세대정보통신연구소/
정보통신학과 조교수
<주관심분야 : 통신시스템, 컴퓨터>



김현덕(정회원)
1997년 경북대학교 전자공학과
학사 졸업
1999년 한국과학기술원
전기 및 전자공학과 석사
2002년 한국과학기술원
전자전산학과 박사
2002년~2003년 노베라옵틱스코리아 선임연구원
2003년~현재 경북대학교 전자전기컴퓨터학부
전임강사
<주관심분야 : 통신시스템, 광통신, 광네트워크>