

Remote-Agent를 통한 모바일 File Search 설계 및 구현

곽동규^o, 김상태, 이유근, 박원배

경북대학교 정보통신학과

{kwak^o, storm, cinema74}@inc.knu.ac.kr, wbpark@ee.knu.ac.kr

Design and Implementation of Mobile File Search throughout Remote-Agent

Dongkyu Kwak^o, Sangtae Kim, Yougeun Lee, Wonbae Park

Dept. of Information and Communications Kyungpook National University

요 약

현재 많은 부분에서 모바일을 이용한 다양한 서비스와 그러한 환경을 지원하기 위한 여러 가지 미들웨어를 최적화 한 기술들이 선을 보이고 있다. 특히, 썬마이크로시스템즈사에서 제안한 차세대 접속 용 네트워크 기술인 Jini 는 실제계의 네트워크에 연결된 주변 기기와 응용 프로그램을 사용하고자 하는 서비스에 등록과 동시에 활용 할 수 있는 기술로서 Desktop 환경에서 지원을 하고 있다. 이러한 지원의 조건에 반하여 본 논문에서는 모바일 환경에 적용할 수 있는 새로운 접근 방법을 모색함으로써 모바일 환경에서 동적으로 원격지의 노드를 찾아낼과 동시에 활성화 된 모바일 디바이스를 편리하게 사용 할 수 있게끔, 리스트 정보를 모바일 디바이스에서 보여주어 사용자로 하여금 언제 어디서나 필요한 문서를 찾아낼 수 있는 모바일 File Search 서비스를 설계 및 구현 한다.

1. 서 론

최근 향상된 모바일 환경과 네트워크 환경을 기반으로 무선 통신의 활용이 상당한 수준에 이르고 있다. 모바일과 무선 컴퓨팅에서 Laptop, handheld PC, wearable 컴퓨터와 같은 휴대용 디바이스 보급 증대로 무선인터넷을 통한 업무수행이 늘어나고 있다. 이렇게 모바일화 됨에 따라 기존의 유선 컴퓨팅 환경에서의 편리함을 무선 환경에서도 동일하게 누릴 수 있는 방법을 모색하여 왔다. 그러나 무선 환경에서의 네트워크 리소스와 애플리케이션들은 모바일 사용자들이 이동할 경우를 모두 수용해 주지 못하게 된다. 이런 문제 해결방법으로 많은 대안들이 연구 되고 있으며, 특히 네트워크 환경에서 유동적인 원격지의 서비스를 이용하기 위한 사용자 중심의 모바일 환경 접속 기술이 점차 부각되고 있는 추세다.

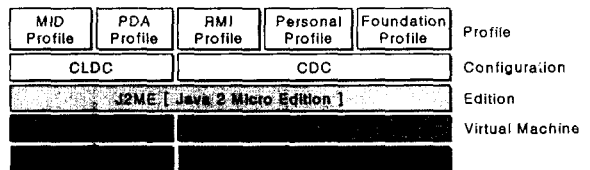
원격지의 서비스를 이용하기 위해 Java 기반의 홈 네트워크기술에서 가장 기본이 되는 프로토콜로서 기존의 지원되는 Discovery를 한 단계 향상시킨 노드 검색과 동시에 가능한 서비스를 이용할 수 있게끔 이루어지도록 한다. 일반적으로 원격지의 애플리케이션을 이용하고자 할 경우, 네트워크 관리측은 만족할 수 있는 인스턴스를 찾을 때까지 목록검색을 계속 시도하게 된다. 이것은 모바일 디바이스 적용을 위해 필요한 컴퓨팅 네트워크 환경을 충분히 지원할 경우이며, 그렇지 않다면 상당히 부적합한 환경으로 되어진다. 즉 어떤 서비스를 이용하기 위한 쿼리를 요청하여 그 결과로 돌아오는 서비스 인스턴스들이 네트워크의 혼란을 초래할 수도 있게 된다[1].

본 논문에서는 이러한 점들을 고려하여 제공 되어지는 서비스 환경의 기반적인 부분을 모바일 환경에 적용함으로써 서비스가 가능하게 하기 위한 Remote-Agent를 통한 모바일 File Search를 설계 및 구현 한다.

2. 관련 기술

2.1 J2ME Platform

J2ME는 제약된 환경 즉 모바일 환경에서 Java 응용 프로그램을 실행하기 위해서 개발된 소형 플랫폼이다. J2ME 프로그래밍이 적용되는 디바이스들은 다시 디바이스들의 특징에 따라 최소한의 사양만을 정의 해 놓은 Configuration (CDC, CLDC)과 그 위에 다양한 디바이스 환경에 적용한 Profiles (MIDP)로 구분된다. 일반 Java 플랫폼에서는 Java 프로그램을 실행시키기 위해 Virtual Machine(VM)으로 JVM을 이용하지만, 그림 1에서는 기존 JVM이 아닌 소형 컴퓨팅 디바이스들을 위한 KVM(Kilobyte VM)과 CVM(Classic VM) 두 개의 Java VM이 J2ME상에서 이용됨을 보여 주고 있다[2].



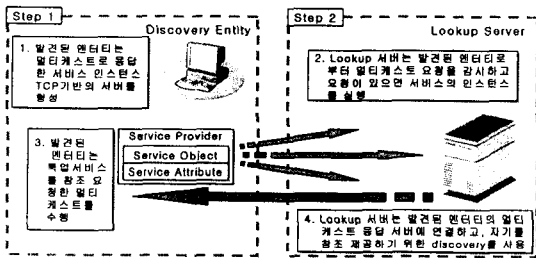
[그림 1] J2ME 기반의 VM 스택

KVM은 16~32bit의 프로세스와 160~512K의 메모리를 가진 디바이스를 위한 VM인 반면 CVM은 32bit 이상의 프로세스와 2MB이상의 메모리를 가진 디바이스에서 사용된다. 이러한 적용환경을 그림 1의 J2ME 스택에 따른 VM의 구분 영역에서 볼 수 있다[3,4].

2.2 Discovery Protocol

Discovery Protocol의 목적은 Lookup 서비스들을 찾고 찾는 서비스로부터 서비스 Proxy를 획득하여, 서비스 Proxy에 정의된 기능을 수행하게 된다. 즉 Proxy들은 Lookup 서비스로부터 얻을 수 있지만 Lookup을 위한 Proxy는 Discovery를 통해 얻게 된다. Discovery는 존재하는 정보를 알지 못해도 향후 시작되는 서비스를 발견하거나, 네트워크상의 Lookup 서비스를 공유하여 자신의 서비스를 등록할 수 있도록 설계되어 있다[5].

Discovery Protocol은 그림 2에서 크게 두 가지 유형으로 나뉜다. 첫째 Step 2에서 Multicast로 네트워크의 애플리케이션을 찾기 전 추가적인 정보 없이 서로를 찾는 자생적인 상호 작용으로 하나의 서비스가 Lookup 서비스를 찾으려고 하거나 Lookup 서비스가 네트워크에서 자신의 존재를 알리려고 할 경우, 둘째, 네트워크에서 Lookup 서비스가 고정된 경우로 Step 1에서 Step2의 Lookup 서비스를 찾아 Lookup 서비스 URL을 사용 명시하는 경우이다. 이런 3가지의 프로토콜(두개의 자생적인 프로토콜과 한 개의 고정 프로토콜)은 Discovery의 핵심이 된다[6].



[그림 2] Discovery Protocol 구성 요소

3. 시스템 구성 및 설계

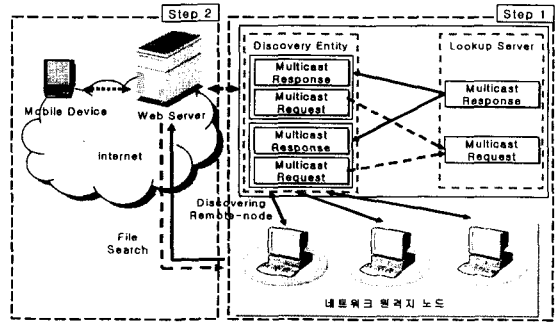
본 논문에서는 Desktop 환경으로 이루어지고 있는 Discovery Protocol을 제한된 모바일 디바이스에 적용하기 위한 하나의 해결책으로 Discovery를 통한 원격지의 노드를 Web 서버를 이용하여 Lookup 서버에 등록하고 자동 검색하는 Remote-Agent를 고안하여 서비스를 모바일 환경에서 이용할 수 있도록 설계 및 구현 한다.

3.1 시스템 구성

Web Server를 기반으로 본 논문에서 제시한 모바일 환경을 지원하기 위한 Remote-Agent를 통한 시스템 구성도는 그림 3과 같다

그림 3구성도에서 Remote-Agent를 통한 File Search Service는 크게 두 가지로 구분된다. Step 1에서 Lookup 서비스에 등록되어 있는 Service 객체들 중 원하는 File Search Service를 Multicast를 통해 요청하고 응답함에 따라 노드 목록을 검색하여 선택한 Service 복사본인 Proxy를 다운로드하여 Service를 이용하는 부분, Step 2의 무선 인터넷상에서 Jini Service를 이용하기 위해서 필요한 Web Server를 통한 Lookup 서비스로부터

받은 Proxy를 처리해 주는 부분으로 이루어진다.



[그림 3] Remote-Agent 시스템 구성도

Mobile Device와의 연동을 위하여 자동으로 원격지의 노드를 검색하고 검색된 노드를 이용가능하게 하는 Remote-Agent를 이용하여 모바일에서 File 검색 및 제어가 이루어진다.

3.2 시스템 동작 과정

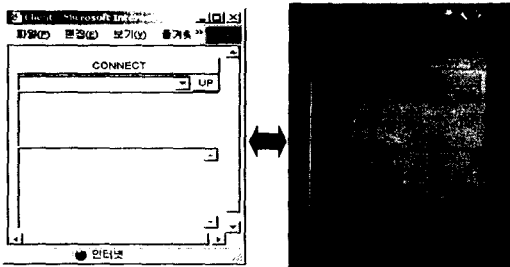
우선 서비스 제공자는 Lookup 서버의 위치를 찾기 위해서 Unicast나 Multicast를 이용하여 검색을 하게 된다. Lookup 서버의 위치를 확인한 후 서비스 제공자는 서비스 등록을 요청한다. 그리고 Discovery 프로세스를 제어하는 가장 편리한 방법으로 LookupDiscoveryManager를 사용하는데 이 클래스는 Unicast나 Multicast의 Discovery 프로토콜을 제어하는 모든 기능과 리스너 등록 및 제어기능을 하며 요청에 대한 하나의 자바 객체인 registrar를 서비스 제공자에게 보낸다[7].

서비스 제공자는 LookupDiscoveryManager에서 보내어진 registrar를 통해서 서비스 Proxy를 보내고 Lookup 서버에 등록하게 된다. 사용자의 메시지를 수신한 Lookup 서비스는 이용자에게 접근할 수 있도록 registrar를 서비스 이용자 쪽에 보내게 된다. 이렇게 연결이 설정되면, 이용자는 구체적인 서비스를 검색하게 되고 선택한 복사본인 Proxy를 다운로드하여 서비스를 이용한다. 또한 무선인터넷을 통한 desktop 환경의 서비스를 사용하기 위해 web 서버에 서비스 검색을 통한 연결을 설정하게 된다. 이렇게 작업이 진행 되고 나면 Remote-Agent는 등록된 사용가능한 서비스 목록을 Service Provider의 노드에서 자동으로 검색하여 Lookup 서버로 Proxy 값을 가져오게 된다. 동시에 그 요청에 의해 LookupDiscoveryManager는 Proxy를 다운로드하여 네트워크상에 존재하는 원격지의 노드를 모바일 디바이스 상에서 찾아 Remote-Agent의 주 역할인 원격지의 서비스 노드를 자동 검색하고 보여 주게 된다. 자동으로 Remote-Agent를 통해 네트워크상에 찾은 등록된 노드 List들은 접속을 위해 J2ME 기반의 웹 브라우저 무선 단말기를 이용, 서비스 작동 사이트의 URL을 통해서 Midlet을 위한 별도의 클라이언트 프로그램을 다운로드 설치할 필요 없이 모바일 File Search 서비스를 이용할 수 있게 된다.

3.3 시스템 구현

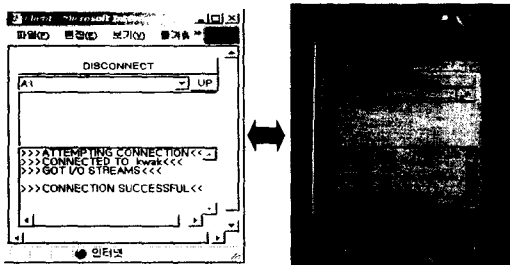
모바일 환경에서 다양한 서비스를 위한 Web Server를 이용 Remote-Agent를 통한 File Search 서비스를 PDA 상의 웹 브라우저를 사용하여 구현된 실행과정은 다음과 같다.

① 그림 4는 연결을 위한 대기 상태에서 실제 PDA 상의 웹 브라우저를 이용하여 Web Server의 URL로 연결을 시도하게 된다. 여기서 Remote-Agent를 통해 원격지 노드를 자동으로 찾고 File Search를 위한 상단의 CONNECT를 누르는 순간 원격지 노드와 연결 된다.



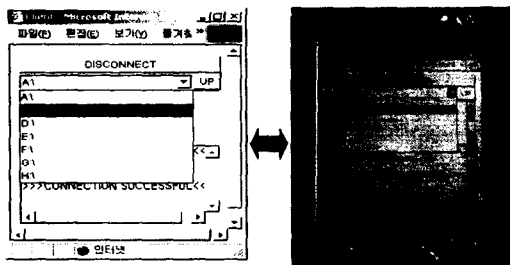
[그림 4] 연결을 위한 대기 상태

② 그림 5는 CONNECT 되어진 상태로 Remote-Agent를 통해 원격지 노드의 Root Directory를 보여 주게 된다.



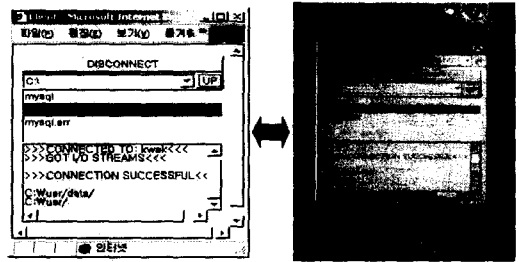
[그림 5] 원격지와 연결된 상태

③ 그림 6은 원격지의 찾고자 하는 Directory를 선택할 경우 그 하위의 Subdirectory를 보여 주게 되며, 이러한 연결 상태 정보를 상태 정보 창에서 직접 보여 지게 한다.



[그림 6] 원격지 컴퓨터의 다른 Directory를 찾는 화면

④ 그림 7은 원격지 노드와 연결된 상태에서의 C:\w\ 하위 Directory를 검색하여 특정 File의 내용을 보여 주고 있다. 원격지 노드에서 사용자가 찾고자 하는 File을 선택 하였을 경우, 그 File에 대한 상태 정보(File 이름, File 크기, File 생성 일자)를 보여주고 있다.



[그림 7] 원격지 노드의 찾은 File에 대한 상태 정보화면

4. 결 론

File Search System은 무선 인터넷 환경에서의 모바일 Device를 위한 Web Server기반의 Remote-Agent를 설계 및 구현함으로써 무선 환경의 분산 패러다임을 가능하게 만들며, 모바일 사용자들에게도 유선 환경에서의 Service를 보다 효율적으로 사용할 수 있는 방안을 제시하고 있다.

향후 이러한 분산 네트워크 접속 기술을 이용하여 모바일에서의 원격기기제어, 정보가전들에 대한 애플리케이션 개발과 Mobile Device에 대한 다양한 Service 등을 통한 실생활에서의 적용을 위해 많은 연구를 계속해야 할 것이다.

5. 참고 문헌

- [1] Choonhwa Lee and Sumi Helal, "An Approach to Enable Context-awareness for Service Discovery", IEEE Computer Society Saint 2003
- [2] Sun Microsystems, "Java™ 2 Platform, Micro Edition (J2ME™) Connected Device Configuration (CDC)", <http://www.sun.com/software/j2me>
- [3] Sun Microsystems, "Java 2 Platform Micro Edition", <http://java.sun.com/j2me/index.jsp>
- [4] Sun Microsystems, "CDC: An Application Framework for Personal Mobile Devices", Java™ 2 Platform, Micro Edition(J2ME) June, 2003, <http://www.sun.com/software>
- [5] Scott Oaks, Henry Wong, "Jini in a Nutshell," O'Reilly Press, 2000
- [6] W.K. Edward, "Core Jini 2E", Prentice hall, 2000
- [7] PsiNaptic, "A Jini™ Lookup Service for Resource-constrained Devices", IEEE International Workshop on Networked Appliances January 15-16, 2002