

지니를 이용한 웹 기반 프린팅 시스템 구현

이수영, 김형길, 김명균, 허정석
울산대학교 컴퓨터·정보통신공학부

Implementation of Web-based Printing System Using Jini

Su-Young Lee, Hyoung-Gil Kim, Myung-Kyun Kim, Jeong-Seok Heo
School of Computer Engineering and Information Technology, University Of Ulsan

요 약

WWW(World Wide Web)은 플랫폼에 독립적인 특성과 인터넷의 보편화로 가장 일반적인 사용자 인터페이스가 되었다. 본 논문에서는 이러한 특징을 가지는 웹과 새로운 분산 기술인 지니 기술을 접목시켜 웹 기반의 프린팅 시스템을 구현하였다. 이 시스템은 Local Area Network 내에서 프린트 서비스가 네트워크에 접속되어 있으면 프린터 드라이버를 추가하지 않고 어디서나 사용 가능하며, 지니 기술의 Plug&Work 개념을 적용하여 웹 브라우저를 통해 네트워크에 연결된 각 서비스들을 검색, 이용할 수 있도록 하였다.

1. 서론

최근 World Wide Web을 이용한 기술들이 소개되고 있으며, 웹 기반의 네트워크 관리 응용 프로그램들 그 예로 들 수 있다. 웹 기반을 이용한 기술은 이미 오래 전부터 연구되어 왔기 때문에 웹 기술을 적용하기 위해 투자해야 할 시간과 비용이 적게 든다는 장점이 있다.

지니 기술을 이용한 웹 기반 프린팅 시스템은 웹이 가지는 장점과 지니 기술이 가지는 분산 환경의 기능을 동시에 만족시킨다. 디지털 카메라, 셀룰러 폰, 가정 기기 등 마이크로 프로세서가 사용되어지는 모든 장치들이 네트워크에 접속할 수 있는 형태로 변해 가고 있는 추세에서 복잡한 네트워크 기반 환경을 관리하는 것은 매우 어렵다. 이런 복잡한 네트워크 환경을 쉽게 관리, 사용할 수 있는 연구가 현재 진행되고 있으며 대표적으로 선 마이크로시스템즈의 지니 기술이다. 이 기술은 단순히 네트워크에 연결되지만 하면 다른 부가적인 동작없이 서비스를 원하는 클라이언트는 누구나 이용할 수 있으며 순수 자바로 구현되어 있기 때문에 자바 코드만 실행할 수 있는 자바 가상 머신(JVM)만 있으면 된다.[1,2,4,6]

즉 지니 연합체(Community)를 구성하기 위해서는 먼저 장치 내부에 자바 가상 머신이 필요하며, 각 네트

워크 장치들은 지니 코드를 장착해야 한다. 지니 코드를 구동시킨 후의 과정은 자바로 구현해야 한다. 본 논문에서는 이러한 지니 기능을 바탕으로 한 웹 기반 프린팅 서비스 시스템 구현에 대해 기술하고자 한다.

2장에서 관련연구로 지니 기술의 구조와 특징에 대해 기술하고, 3장에서는 본 논문에서 구현한 웹 기반의 지니 프린팅 서비스 시스템에 관해 설명한다. 끝으로 4장에서 결론 및 향후 연구 방향을 기술한다.

2. 관련 연구

웹 기반의 지니 프린팅 시스템을 이해하기 위해서 먼저 지니 기술의 구조와 동작에 대해 살펴보도록 하겠다.

지니 기술은 SUN에서 제안하고 있는 기술로서 네트워크 상에서 서비스를 제공하고, 서비스를 이용하고자하는 프로그램들 사이의 상호작용을 생성해 내기 위한 하부구조를 제공한다.[1,4] 지니 기술은 서버-클라이언트 개념이 아닌 서비스-사용자의 개념으로 이해해야 한다. 서비스는 네트워크에 접속 또는 접속 해제가 가능하고, 사용자 즉 클라이언트는 접속되어 있는 서비스를 이용하는 형태이다.[1,7] 사용자는 서비스를 이용하기 위해서는 자바 객체(Proxy Object)를 통해서만 가능하다.[1,4,6,7]

이런 지니 기술의 목적은 네트워크 Plug&Work, 하드웨어와 소프트웨어의 경계선을 제거하고, 원격 서비스를 기반으로 하는 구조를 가지는 것이다.[1,4,7] 즉 하드웨어나 소프트웨어에 관계없이 지니를 탑재한 디바이스가 네트워크에 연결되면 그 디바이스에 대한 일련의 조작이 가능하다. 지니 시스템은 그림 1과 같이 서비스 이용자 (Jini Client), 서비스 제공자(Jini Service), 서비스 관리자(Lookup Service)로 구성되며, discovery, join, lookup 프로토콜을 통해서 서로 상호작용하며 RMI(Remote Method Invocation)를 통해서 원격 객체를 전달한다.[1,3,6]

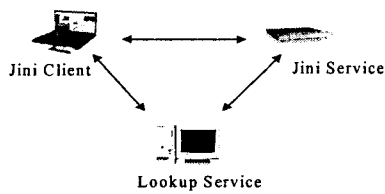


그림 1 지니 시스템 구조

지니 시스템에서 서비스는 서비스와 상호작용을 위해, 사용자와 룩업 서비스에 자신의 존재를 알리기 위해 discovery, join, lookup 이라는 프로토콜을 사용하며, 위 3개의 프로토콜이 지니 시스템의 중심을 이룬다.[1,4,6,7]

- Discovery : 디바이스가 네트워크에 접속한 후 서비스를 등록하기 위해 lookup service를 찾을 때 사용되는 프로토콜이다. 종류에는 Unicast, Multicast의 두 가지 종류가 있다.

- Join : 서비스가 lookup service에 자신의 원격 객체를 등록하고, 자신의 서비스를 제공하기 위해 이용되는 프로토콜이다.

- Lookup : 서비스 이용자가 lookup service에 등록되어 있는 서비스를 찾거나 호출할 때 이용되는 프로토콜이다.

지니 기술은 분산환경이라는 점에서는 CORBA와 유사하다. CORBA는 분산환경에서 응용 프로그램들을 쉽게 개발할 수 있는 API를 제공하며, 망관리 응용과 같은 분산 응용 프로그램들 사이의 상호 연동을 쉽게 해 주고, 통신망 관리나 서비스 관리 문제를 수월하게 해결할 수 있다. 이러한 CORBA와 지니의 차이점은 CORBA는 자바나 C++와 같은 언어로 구현가능하고, 서로 연동이 가능하다. 즉 언어에 독립적이다. 하지만 지니는 순수 자바로 제작된 기술이기 때문에 언어 종속적이다. CORBA의 ORB와 지니의 lookup service는

유사한 기능을 하지만, 지니의 lookup service는 서비스 제공자와 이용자 모두 서로간의 위치에 대한 데이터가 없어도 동작이 가능하다는 것이 장점이다.

3. 지니를 이용한 웹 기반 프린팅 시스템

웹을 기반으로 하는 지니 프린팅 시스템은 웹의 특징을 포함하기 때문에 서비스 제공자와 서비스 이용자사이의 인터페이스만 정의되면 쉽게 이용할 수 있다는 장점이 있으며, 많은 비용을 들이지 않고 관리할 수 있다. 만약 애플리케이션으로 제작할 경우 구현 환경에 따라 새로운 인터페이스를 습득해야 하지만 웹 기반으로 하는 경우 많은 사람들에게 친숙한 웹 브라우저를 사용하므로 이용 및 관리가 쉽다.

3.1 전체 시스템 구조

구현 환경은 자바 가상 머신이 설치되어 있고, 지니 코드를 실행할 수 있는 API가 존재하면 어떤 플랫폼에서도 동작이 가능하다.

그림 2는 웹 기반 지니 프린팅 시스템의 구조이다.

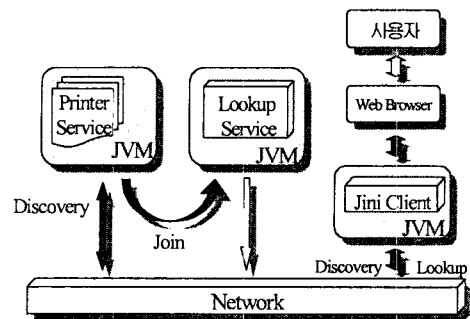


그림 2. 웹 기반 지니 프린팅 시스템 구조

- ▶ Printer Service : Discovery, Join 프로토콜을 이용하여 lookup service를 찾고 등록된 후, 사용자로부터 일반 텍스트 문서의 프린트 요청이 전송되어 오면 사용자로부터 넘겨받은 문서를 프린터로 출력한다.

- ▶ Lookup Service : 네트워크에서 독립된 호스트에서 실행되며 Jini Client에게 Printer Service에 대한 정보를 넘겨주는 역할을 한다.

- ▶ Jini Client : Discovery, Lookup 프로토콜을 사용하여 프린트 서비스를 찾고 다운로드 한 후, 사용자는 웹 브라우저를 이용하여 문서를 프린터로 출력할 수 있다.

3.2 프린팅 시스템 동작 알고리즘

프린팅 시스템의 동작은 다음과 같다. 모든 일은 lookup service를 통해서 일어나기 때문에 lookup service가 먼저 구동되어야 한다. lookup service는 별도의 호스트에서 코드를 실행하면 된다. 다음으로 서비스를 제공하고자 하는 프린트 서비스가 준비되어야 한다. 과정은 discovery 프로토콜을 사용하여 lookup service를 찾고, join 프로토콜을 사용하여 자신의 원격 객체를 lookup service에 등록한다. lookup service에 등록된 원격 객체를 통해서 사용자는 프린트 서비스에 접속이 가능하다. 프린트 서비스가 준비되면 서비스 제공을 위한 준비는 끝나고, 사용자 코드를 구동시켜야 한다. 사용자도 마찬가지로 discovery 프로토콜을 사용하여 lookup service를 찾고, 등록되어 있는 원격 객체를 다운로드 한다. 다운로드 한 원격 객체를 통해서 사용자는 웹 브라우저를 이용해 프린트 서비스에 접속하고, 프린트 서비스를 이용할 수 있게 된다.

3.3 시스템 동작 구현

그림 3은 lookup service가 구동되는 것으로 프린트 서비스와 사용자를 연결시켜주는 고리역할을 한다. lookup service는 원격 객체를 다운로드하기 위해 웹 서버가 구동되어야 하고, RMI(RMI Activation Daemon)는 원격 객체를 호출하기 위해 필요하다. 서비스 이용자는 RMI를 이용해서 원격 객체를 호출하기 때문에 꼭 필요한 서비스이다. 요약하면, 웹 서버 - RMI - lookup service의 순서로 구동된다.

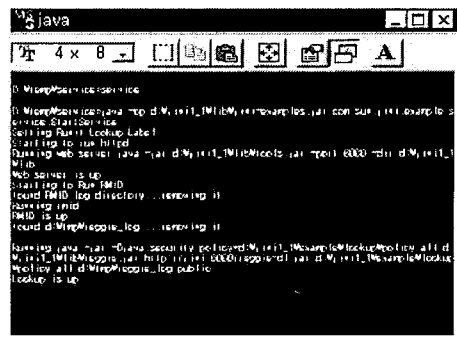


그림 3. lookup service 실행 화면

다음으로 프린터 서비스가 구동되어야 한다. 프린트 서비스는 lookup service를 검색, 발견하고 원격 객체

를 등록하기 위해 discovery, join 프로토콜을 사용한다. 그림 4는 프린트 서비스가 구동되는 화면이다.

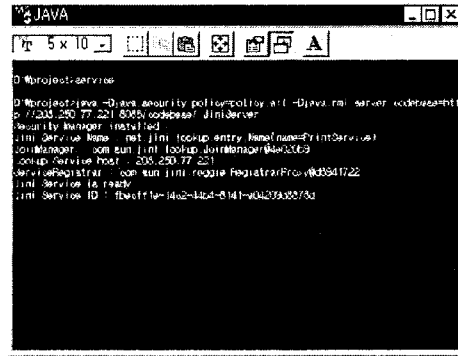


그림 4. 프린터 서비스 실행 화면

위의 과정으로 프린트 서비스를 이용할 수 있는 환경이 갖추어 지게 된다. 프린트 서비스를 이용하기 위해서는 사용자 호스트에서 서비스를 이용하기 위해 정의된 인터페이스를 웹 브라우저로 실행하면 된다. 사용자 코드가 구동되면 lookup service를 검색, 발견하고, 원하는 서비스의 원격 객체를 다운로드 한 후 RMI를 통해서 프린트 서비스의 method를 호출함으로써 최종적으로 서비스를 이용한다. 그림 5는 사용자의 호스트에서 웹 브라우저를 이용한 서비스 사용 화면이다.

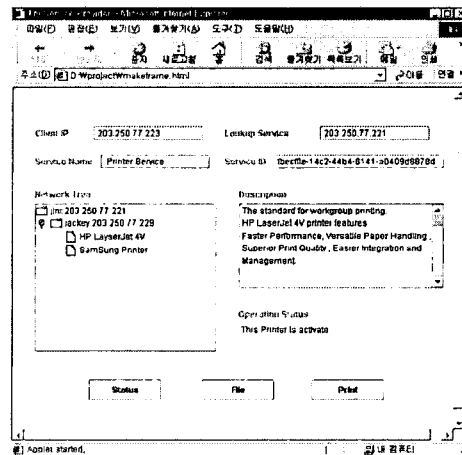


그림 5. 웹 기반 지니 클라이언트 실행 화면

서비스 사용방법에 대해 간략하게 설명하면 다음과 같다.

- ▶ Client IP : 서비스 사용자 호스트 주소.
- ▶ Lookup Service : lookup service가 실행되는 호스

트 주소.

▶ Service Name : 프린트 서비스가 제공하는 서비스 이름.

▶ Service ID : 서비스 식별자.

▶ Network Tree : lookup service와 프린트 서비스의 네트워크 연결 상태.

▶ Description : 서비스 설명.

▶ Operation Status : 서비스 동작 상태.

Status 버튼은 프린트 서비스의 현재 상태를 얻어 올 경우 이용하는 것이고, File 버튼은 출력하고자하는 문서를 선택하는 버튼이다. 마지막으로 Print 버튼은 사용자의 웹 브라우저에서 이 버튼을 누르게 되면 서비스를 제공하는 호스트와 연결된 프린터에서 문서의 내용이 출력된다.

Services in a Flexible Network, " IEEE 1999

[5] Merlin Hughes, Michael Shoffner, Derek Harner

"Java Network Programming," MANNING 1999

[6] W.Keith Edwards, "Core Jini," Prentice Hall 1999

[7] SUN Microsystems Jini Connection Technology

"http://www.sun.com/jini/index.html"

[8] The Community Resource For Jini Technology

"http://www.jini.org"

4. 결론

프린트 서비스에서 지니 기술의 기본 목적으로 본다면 프린터만 네트워크에 접속되어야 하지만 아직 프린터에 지니 코드를 탑재할 기술이나 환경이 준비가 되지 못한다. 본 논문에서 구현한 프린트 서비스에서는 임의의 호스트에 프린터가 프린터 포트로 접속되어 있는 환경에서 서비스를 구현하였다. 앞으로 하드웨어 기술과 지니, 자바의 기술이 향상된다면 하드웨어에 직접 지니 코드를 탑재할 수 있을 것이고, 그 결과 지니 탑재 하드웨어는 네트워크에 접속만 되면 어디서든 이용 가능하게 될 것이다. 자바 기술의 제한으로 프린트 서비스를 실행한 결과 프린트가 진행되는 속도가 느린 것이 문제다. 비주얼 계열 프로그램의 경우 윈도우에서 제공하는 컴포넌트를 가져와서 사용하기 때문에 빠르지만, 지니의 경우 자바의 PrintJob 객체를 импорт해서 프린트 환경을 구현하기 때문에 속도가 느리다. 이 문제는 자바의 기술이 좀더 개선되면 해결될 것이다.

[참고문헌]

[1] Arnold, O'Sullivan, Scheifler, Waldo, Wollrath, "The Jini Specification," Addison Wesley 3rd November 1999

[2] Erik Guttman, James Kempf Automatic Discovery Of Thin Servers : "SLP, Jini and the SLP-Jini Bridge," Sun Microsystems

[3] Jeff Nelson "Programming Mobile Objects with JAVA," WILEY 1999

[4] Ken Arnold, The Jini Architecture : "Dynamic