

PACE UNIVERSITY

Thesis
Master of Science in Computer Science



Client-Server Applications in Java

by
Jasmine J. Ahuja

Advisor
Dr. Howard Blum

December 1997

CONTENTS

1. Introduction
 2. Overview of Concepts
 - 2.1 Distributed Applications
 - 2.2 The Java Language
 - 2.3 World Wide Web
 - 2.4 Client-Server
 3. Java vs. C++
 4. Communication between Client and Server using sockets and Java Server application
 - 4.1 Client as Java Application (Non Web)
 - 4.2 Client as Web Browser (using Telnet Applet)
 - 4.3 Client as Web Browser (using Java Applet)
 5. Sample Applications
 - Client as Java Application (Non Web)
 - Client as Web Browser (using Telnet Applet)
 - Client as Web Browser (using Java Applet)
 6. Some other methods of communication between Client and Server
 - 5.1 CGI Scripts
 - 5.2 Remote Method Invocation
 7. Miscellaneous Concerns
 8. Recommendations for future studies
 9. Bibliography
- Appendix
- A. Code for Version 1- Client as a Java Application
 - A.1 Client
 - A.2 Server
 - B. Code for Version 2- Client as a Web Browser (Telnet Applet)
 - B.1 Client
 - B.2 Server
 - C. Code for Version 3- Client as a Web Browser (Java Applet)
 - C.1 Client
 - C.2 Server

Client-Server Applications in Java

1. Introduction

During the first two decades of their existence, computer systems were highly centralized. A computer was usually placed within a large room and the information to be processed had to be taken to it. This had two major flaws, a) the concept of a single large computer doing all the work and b) the idea of users bringing work to the computer instead of bringing the computer to the user.

This was followed by 'stand alone PCs' where the complete application had to be loaded on to a single machine. Each user had his/her own copy of the software. The major problems were a) sharing information and b) redundancy.

These two concepts are now being balanced by a new concept called computer networks. In computer networking a large number of separate but interconnected computers work together. An application that requires two or more computers on the network is called a network application. The client-server model is a standard model for network applications. A server is a process that is waiting to be contacted by a client process so that the server can do something for it. A client is a process that sends a request to the server. The idea behind a network application is to offload some utilities like the Graphical User Interface (GUI) and in some cases, some processing, from the server to the client. As a result the GUI can be elaborate, thus making the application more user friendly. The other issues involved in client-servers are inter-process communication, security, maintenance and requirement for special software to run the client.

In a 'conventional' client-server model, a copy of the client process lies on every client machine and the client sends a request to the server through a socket to the port that the server is listening on. This model is popular as it offers improved graphical user interface. However, it has the following disadvantages: a) a copy of the client process has to be installed on every user's machine and b) any modification or update to the client process needs to be distributed and installed on every client machine.

In a new, modified view, there is no preinstalled client on the client machine. The client process is loaded onto the client machine each and every time the user initiates a request through his/her web browser using the server's URL/IP address. This 'web oriented view' does not require the user to have a special software to run the application. Moreover maintenance is much easier. The server's administrator can ensure that the users always download the latest version of the client.

In this paper, attempt has been made to study the basic issues and requirements for design and development of web oriented client-server applications and to examine the Java language in the context of these requirements. This paper especially concentrates on the inter-process communication and the user interface.

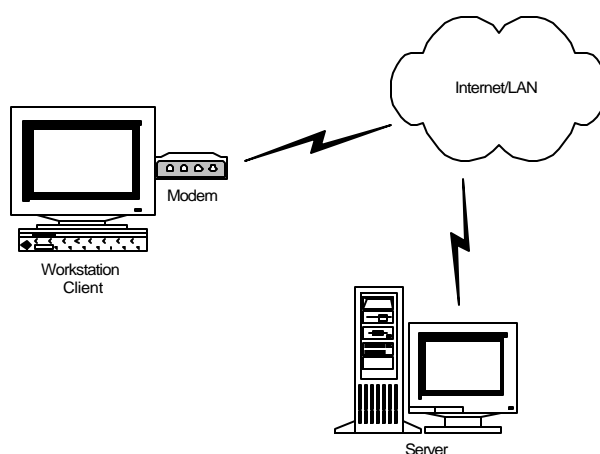
The paper starts with an overview of concepts commonly used in networking. It goes on to discuss communications using sockets in detail and three alternative Java based examples of client-server implementation. The first example is a conventional client-server in which the client as well as the server are applications. The second example is web based, where the client is the web browser and the server is an application and they use HTML and a Telnet applet to communicate. The third example is also web based, but uses a Java applet along with the HTML. The paper also discusses other methods of communication like Remote Method Invocation and CGI scripts in brief. Finally a discussion on applet security is followed by ideas for further studies in this field.

2. Overview of Concepts

2.1 Distributed Applications

The term *distributed applications*, is used for applications that require two or more autonomous computers or processes to cooperate in order to run them. Thus, the distributed system considered in this paper, involves three resources, processing, data and user interface. Both processing and data can be distributed over many computers. The user interface is usually local to the user so that the graphical interface, which consumes high bandwidth, does not have to be transmitted from one location to another.

A Typical Distributed Application Scenario



In distributed computing, the computer network is used to support the execution of program units, called processes, that cooperate with one another to work towards a common goal. This approach has become popular due to a number of developments like:

- ✓ Increase in the number of personal computers
- ✓ Low cost of establishing computer networks with the advancement of technology
- ✓ Computer manufacturers now offer networking software as a part of the basic operating system
- ✓ Computer networks are now an established way of disseminating information

2.2 The Java Language [22]

Java is a new programming language invented by Sun Microsystems. Sun's goal was to allow programmers to create one copy of a program that users could run on almost any computer and operating system. This capability was designed to make Java a vital component of programmability on the Web.

The Java Language (contd.)

Programs written in Java are translated into their own native architecture, in a format called bytecodes. In order to run a Java program, the user needs another program that can interpret the Java program and provide it with the environment and services it needs. The software layer, the Java Virtual Machine (JVM), makes just about any hardware and software platform look the same to the Java program. In effect, a JVM is a device driver for Java programs. Running a Java program with a JVM is currently slower than running an equivalent C program, but JVM technology is improving at a rapid pace.

Java does not support pointers to memory and so Java programs cannot corrupt a computer's memory. It also ensures all array and string references are within each item's bound. It performs automatic garbage collection, which frees up unused memory, so Java programs cannot leak resources.

Java's potential of platform-independent bytecodes and a language designed to go hand in hand with the Internet is enormous. However, it is still a new language and has not been tested on very large projects where applications can migrate over many platforms. It is only when its tools are tested thoroughly will businesses be willing to adopt it as 'the' programming language of the future.

2.3 World Wide Web [15]

The World Wide Web can be briefly described as a global, interactive, dynamic, cross-platform, distributed, graphical hypertext information system that runs over the Internet. The primary concept behind the World Wide Web is hypertext – i.e. instead of reading text in a rigid, linear structure (e.g. a book), the user can easily skip from one point to another using available links. The Web is not limited to any one kind of machine, it can be accessed using an application called a browser like Netscape's Navigator. It is successful in providing so much information because information is distributed globally across thousands of sites. The users do not have to install anything, they can go to a particular site to view information and when they are done their system reclaims the disk space. Information on the Web can be updated any time as it is contained on the site that published it and thus is accessible to the administrator of the site.

No single entity "owns" the World Wide Web. Given the enormous number of independent sites that supply information to the Web, it is impossible for any single organization to set rules or guidelines. There is however, a World Wide Web (W3) Consortium, based at MIT (USA) and INRIA (Europe). The Consortium is a congregation of individuals and organizations interested in supporting and defining the languages and protocols that make the Web (HTTP, HTML etc). The W3 Consortium also provides products (browsers, server and so on) that are freely available to anyone who wants to use them.

2.4 Client-Server

The client-server model is a standard model for network applications. A server is a process that is continuously running and waiting to be contacted by a client process. A client process initiates contact with the server by connecting to it at a specified port.

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.