Assignment #1: Implementing Distributed Systems with Java RMI and CORBA

January 28, 2007

1 Overview

The purpose of this assignment is to increase your familiarity with Java RMI and CORBA, and to give you a chance to gain some implementation experience working with these technologies. The assignment consists of two parts, each worth 100 points, concerning Java RMI and CORBA, respectively. Only the first part of the assignment is required. You may, however, choose to complete the second part for extra credit. The assignment will be graded out of 100 points; it is possible to receive 200 points. This gives you an opportunity to earn significant extra credit that might later be applied to more challenging assignments.

2 Part 1: A Java RMI Chat System

The first component of the assignment requires you to develop a basic internet chat system using Java RMI. Your system will enable two users to exchange text messages over the internet using a simple graphical interface. You may choose to implement your system using a client-server architecture similar to the example presented in class, or you may choose to use a peer-to-peer architecture. My suggestion would be to use the latter since it will simplify your system design.

Your system will provide a basic user interface consisting of two text areas. The first text area will be used to enter text to be *sent* to the remote client. You may choose any transmission strategy you wish, though it may be most convenient to send text one line at a time, and to clear the entry area after each transmission. The second text area will display text *received* from the remote client, as well as text *sent* by the local client. The second text area will be read-only.

Your system will make no assumptions about the client end-points. In other words, the client application will enable users to identify the address of the remote chat participant.

I have intentionally specified few design details. Your focus should be on *functionality*. Don't waste your time adding bells and whistles, picking fancy icons, etc. I will focus on your use of RMI in grading this portion of the assignment.

3 Part 2: A CORBA Chat System

The second component of the assignment is analogous to the first, but requires you to implement your chat system using CORBA. More specifically, your system will provide one user interface implemented in Java, and one user interface implemented in C++. The Java interface will be identical in presentation to the interface developed for the first part of the assignment. The C++ interface will be less sophisticated; it will provide a simple command-line interface.

Sun provides an ORB with CORBA bindings for Java. Orbacus provides an ORB with CORBA bindings for C++[1]. You may, however, choose to use any ORB you wish.

Again, you should focus as little as possible on presentation. I will grade this portion of the assignment based on your use of CORBA.

4 Hand-In Instructions

The first component of the assignment is due February 1st. You are required to submit a *zip* file containing all of the relevant source code **before class**, and to provide me with hard copies of your source listings **at the start** of class. (Please remove the ".zip" file extension when submitting your files to prevent the mail server from stripping the attachments.) You will also be asked to schedule a time for a brief demonstration.

The second component of the assignment, if you choose to complete it, is due February 6th. The same assignment deliverables and associated hand-in schedule apply.

5 Notes on Collaboration

You may work individually or in groups of two. **Aside from your partner, please do not consult *anyone* other than me on *any* aspect of this assignment.**

[1] The Orbacus ORB implementation is free for academic use. Visit their website for download details.