

# Course ISE 431: Distributed Information Systems

## Recitation 5 Exercise

Michael J. May

November 17, 2009

This recitation will introduce the Java RMI model for Remote Process Invocation. The Java model is based on the client stub/server stub model that we talked about in class.

### 1 Hello Example

As a simple example, let us work over the major parts of a trivial RPC RMI example:

- The remote class interface - it must extend the class `Remote` to indicate that it will be used for RPC
- One or more remote classes which implement the interface. In this example it will extend `UnicastRemoteObject` (so it doesn't do multicast). It must implement the functions defined in the remote class interface.
- The server which offers the remote object to others to use. The server holds on to the object and publishes its availability using the **rebind** function and URL. The function is similar to the BSD `bind` command.
- A client which knows where the server is located and the name of the remote object that it wants. It pulls the remote object using **lookup**, casts it to the class that it wants, and can use it.

Understandably, the client program must be aware of the interface that the remote object implements. That implies that it is compiled with it (or uses Reflection, but that's a different story).

Compile the code using **javac** and the remote object implementation with **rmic**. The second compiler is needed to create the stub which is used to handle the network communication.

In this recitation, the students did the following steps:

- Download the prepared Hello code and ran it on a single computer
- Modify the Hello code to send a string to the server and receive a Hello string in return which includes the original one.
- Modify the client to access the Hello server on a different computer in the computer lab.

In the next session I wish to have them to do the bank accounts example.