Continuous Assessment 5

Solution Adopted

Mario Ríos Muñoz
9 April 2014
Computer Science Degree
Advanced Programming Course

Introduction:

The idea is to build a simple application using a client-server architecture, to control the 'Colours' class created during the previous lab sessions. The server will be in charge of executing this Colours class, and will stay listening for the command of the ControlModule (the client), which will allow stoping & resuming the execution of the Colours classs.

Two versions of the solution have been submitted, one implementing the Client-Server communications with sockets, and another one using RMI.

Sockets solution:

For this solution, we instantiate a socket, with an specific port number (common for both server & client) so that the server is always listening that port, and the client eventually sends a message through that socket. The message sent is a String containing the desired action (in this case "stop" or "resume") so, when the server receives the string, parses it and perform one operation or the other over the Colours class (basically, if the string is "stop" we close the gateway, and otherwise, we open it.

A problem with this kind of solution is that the port must be pre-established (for both sides) and given the case that it were already being used by another process, we would get an exception.

RMI solution:

RMI solves the socket implementation problem. For implementing an RMI solution, we must create an object that wraps the functions needed by the client and the server. This class (RemoteControl) implements the ControlModuleInterface interface, an is the one registered by the server in the RMI registry, so that the client can find it. When instantiating RemoteControl, Colours class must be passed, so that the remote control object can modify it.

The client therefore just has to look for the RemoteControl object in the RMI registry, and once found call for the proper method of the object.