

Complementos de Telemática I

Lab Assignment 1: CT1-Auction

RMI & RMI-IIOP

Guillermo Vega
October 2010

Abstract

The different sections of this lab assignment deal with the practical use of the concepts reviewed during the course “Complementos de Telemática I” related to Object-Oriented Middleware technologies. More specifically, the scope of the assignment includes Java-RMI (Remote Method Invocation) and Java RMI-IIOP (Remote Method Invocation-Internet Inter-ORB Protocol). The students are expected to develop very simple client/server applications where the difficulties associated to the mentioned technologies, both from the role of developer and system administrator, can be identified.

1 Goals and assignment contents

The goals this assignment wants to achieve are:

- Get an overall understanding of the common steps associated to the development of client/server applications based on Java RMI and Java RMI-IIOP, stressing the differences with respect to the development of centralized applications.
- Assess in practice the main advantages and drawbacks of Java RMI and Java RMI-IIOP.
- Understand the importance of auxiliary support associated to the involved middleware technologies and, more precisely, to the Object Naming Service.

This assignment does not expect the student to tackle the development of complex distributed applications. It just pretends that the student acquires some basic notions on the usage of the involved technologies so that he can cope with increasingly more difficult concepts and techniques in future formation or professional involvements. Therefore, the assignment consists of the development of a very simple client/server application so that the comprehension of concepts is stressed, and not so much the programming of complex application-specific business logic.

Section 2 introduces the functionality of the applications to be developed. Section 3 describes important advices and guidelines for dealing with the assignment tasks when using Java RMI technology, as well as some important questions that the student should be capable of answering after completing those tasks. Section 4 does the same for the case of RMI-IIOP technology. Finally, section 5 enumerates some important remarks with respect to procedural and organizational issues.

2 The CT1-Auction application

The distributed application to develop is intended to provide very simple support to the management of an on-line auction company: CT1-Auction.

The application (server-side) will consist of two distributed functional blocks called *ItemManager* and *AuctionManager*. On the one hand, the *ItemManager* maintains a list of items that can be submitted to an auction. On the other hand, the *AuctionManager* is in charge of the auctions, approving bid requests and checking the closure of auctions.

A user of the distributed application should be allowed to:

- Post an item to the *ItemManager*
- Put an item of the *ItemManager* for sale at an auction
- List the items of the *ItemManager* and their status
- Make a bid for a particular item to the *AuctionManager*
- Check the status of an auction from the *AuctionManager*

3 Development with Java RMI

A new Java SE (Standard Edition) 6 project within the Eclipse Development Environment at `balbas.tel.uva.es` should be created with the name A1-AUCTION-RMI. The starting code provided by the teacher can then be imported into Eclipse for shortening the development process. The teacher will inform on the way for doing such code importing at the lab sessions.

3.1 Development steps

1. Get the starting code from the following path at `balbas.tel.uva.es`:
`/home/labs/ct1/soft/A1code/`
2. Develop the implementation of the interface `a1.rmi.auction.IAuctionManager`
 - a. Use the java classes `a1.rmi.auction.AuctionStatus` and `a1.rmi.auction.Bid` for managing the auctions
 - b. When creating an auction throw the exception `a1.rmi.auction.BadAuctionException` if the auctions parameters are not valid or there is another auction associated to the item
 - c. When validating a bid consider the cases that the item is not at auction, the bid is too low (another offer or the base price is higher) or the auction is closed (throw the specific exceptions in each case)
 - d. Note the *AuctionManager* has to control the duration of an auction (see the class `a1.rmi.auction.AuctionStatus`)
 - e. For keeping the information in of the auctions updated, it is advisable to use the class `java.util.Hashtable` of the Java API (see <http://java.sun.com/javase/6/docs/api/>)
3. Develop the implementation of the interface `a1.rmi.item.IItemManager`
 - a. It is very **important** to consider that the *ItemManager* should access some functionalities offered by the *AuctionManager* for performing some of its tasks
 - b. Use the class `a1.rmi.item.Item` for managing the items

- c. The *ItemManager* has to assign unique integer identifiers for the items. It is recommended to simply use a counter for this issue
 - d. When listing the items, check their auction status with the *AuctionManager*. In addition, use the `getDescription` method in the *Item* class for obtaining the formatted output
 - e. When putting an item at auction, check the existence of the item and throw a `al.rmi.item.NoItemException` when necessary
4. Develop three client applications for accessing the functionality offered by the *ItemManager*. These two applications should be invoked in the following way:
- a. `java al.rmi.clients.PostItem <name> <seller>`
This client application should inform the user about the success of the operation and the assigned identifier in case of success
 - b. `java al.rmi.clients.AuctionItem <itemId> <basePrice> <numberOfMinutes>`
This client application should inform the user about the causes of failure. In addition, it is important to take into account that this client application will only communicate with the *ItemManager*
Note that `basePrice` is the minimum price of the item in order to be sold, while `numberOfMinutes` refers to the duration of the auction proposal
 - c. `java al.rmi.clients.ListItems`
This client application should present the information about the managed items. It is important to take into account that this client application will only communicate with the *ItemManager*
5. Develop a client application for accessing the functionality offered by the *AuctionManager*. This application should be invoked in the following way:
- a. `java al.rmi.clients.MakeBid <itemId> <offer> <buyer>`
This client application should inform the user on whether the bid was approved or not

3.2 Questions

These questions can be used as a set of guidelines for identifying important aspects of the assignment that could be considered for inclusion in the written reports (see section 5)

- What were the most difficult aspects identified during the development process?
- Was this development process too different from the centralized case?
- Where are the stubs and skeletons?¹
- What pieces of code differentiate a RMI application from a centralized one?
- Try to execute the applications in the same machine and then try to distribute them (e.g. test your clients/servers with those developed by other groups!)

¹ Hint: <http://java.sun.com/j2se/1.4.2/docs/guide/rmi/getstart.doc.html> and also <http://java.sun.com/javase/6/docs/technotes/guides/rmi/relnotes.html>

Do clients and server need further information? What is the role of the RMI Registry?²

4 Development with RMI-IIOP

4.1 Development steps

Modify the RMI version of the CT1-Auction application for using RMI-IIOP technology. It is advisable to create a new eclipse project (A1-AUCTION-RMI-IIOP) and change the name of Java packages to `a1.rmiiop`

4.2 Questions

These questions can be used as a set of guidelines for identifying important aspects of the assignment that could be considered for inclusion in the written reports (see section 5)

- What were the most difficult aspects identified during the development process?
- Was this development process too different from the centralized case? And from the RMI case?
- Where are the stubs and skeletons?
- What is the main advantage of this approach? Does this assignment show it?
- Try to execute the applications in the same machine and then try to distribute it (e.g. test your clients/servers with those developed by other groups!) Do clients and server need further information? Who plays the role of the RMI Registry in this case?

5 Procedural and organizational issues

Each student will be assessed by means of a written report and a face-to-face oral review. The written report will have a maximum length of 6 A4 pages (12pt) that will be focused on the problems encountered, as well as on conceptual conclusions (see questions in section 3.2 and 4.2). The report will have to be delivered on 22/Nov/2010 and the oral review will take place at the laboratory on 22/Nov/2010.

² Hint: <http://java.sun.com/javase/6/docs/technotes/guides/rmi/codebase.html> and also <http://java.sun.com/j2se/1.4.2/docs/guide/security/permissions.html>