Problem 2: Distributed Auction Service

Due: 5:00pm, April 25, 2003

I. Overview

The objective of this machine problem is to use CORBA to build a Distributed Auction Service, which allows the buying and selling of individual items, using an *English* auction protocol (increasing price, current price visible to all parties). Optionally, other auction protocols could be supported, such as *Dutch* (the public price is decreased until someone bids) or some variant of sealed-bid (each interested buyer submits one bid, all bids are considered at once, prices are not public, and the bidder making the highest bid would win the object being sold for either the highest price–a *first-price* auction–or the second-highest–a *second price* (also called *Vickrey* auction)). The idea for this project is from Ebay and some course projects in other universities.

II. Project Requirements

This course project consists of three phases: Phase I project aims to develop a single item distributed auction server using CORBA; Phase II project aims to develop a auction server with multiple simultaneous auctions and develop an user interface as an applet which eliminates the need to deploy client code to client machines; and Phase III project aims to extend the multiple auction server of Phase II with support for financial transactions (i.e., a Bank server).

A. Basic requirements

- **Undergraduate students**: Finish Phase I project.
- **Graduate students**: Finish both Phase I and II projects.

B. Extra Credit Opportunities

In addition to the above basic requirements, your team can receive additional credit (**10 points over the total 100 points**) on the project by implementing the following extensions:

- **Undergraduate students**: Finish both Phase I and II projects.
- **Graduate students**: Finish all Phase I, II, and III projects.

Detailed requirements for the three phases of this project are outlined below.

III. Phase I - A Simple Auction Server

The first phase of the project is intended to familiarize you with the tools and with the process of building and deploying a simple distributed CORBA-based application. For this phase you are to develop a single item distributed auction server, where a seller can offer an item to be sold, and bidders can bid on the item. No other items are auctioned while the current auction is active. The auction server will keep track of the highest bidder and accordingly update the item price. At any point, the seller may decide to sell the item, in which case the auction server will be ready for another item to be auctioned by a seller. More specifically, two Java applications are required, a server and a client. Any number of clients may connect to the server. The system must meet the following general requirements:

- There are two user roles: Seller and Bidder. A separate User Interface is required for each role. Only one active auction at a time needs to be supported. At any given time, there may be at most one seller and any number of bidders active with respect to a given auction server. *However, the server must not lose information even if clients are repeatedly stopped and restarted*.
- The Client may be built as a single application that supports both Seller and Bidder interfaces, or two separate applications may be built, one for a Seller and another for a Bidder.
- Errors, server exceptions, and invalid user input shall produce reasonably informative error displays to the user.
- Note that for this phase of the project, a single IDL interface Auction is sufficient. This interface should contain the exception declarations, attribute declarations, and all of the operations interfaces that specify the functionality described for the Seller and the Bidder.
- Client and Server shall be deployed on at least two different machines. It is acceptable for the client to be replicated on additional machines. Client and Server shall not operate out of the same directory. The Client environment shall not contain unnecessary Server components.
- The Seller and Bidder roles should provide the functionality specified by the following use case scenarios (illustrated in Figure 1):

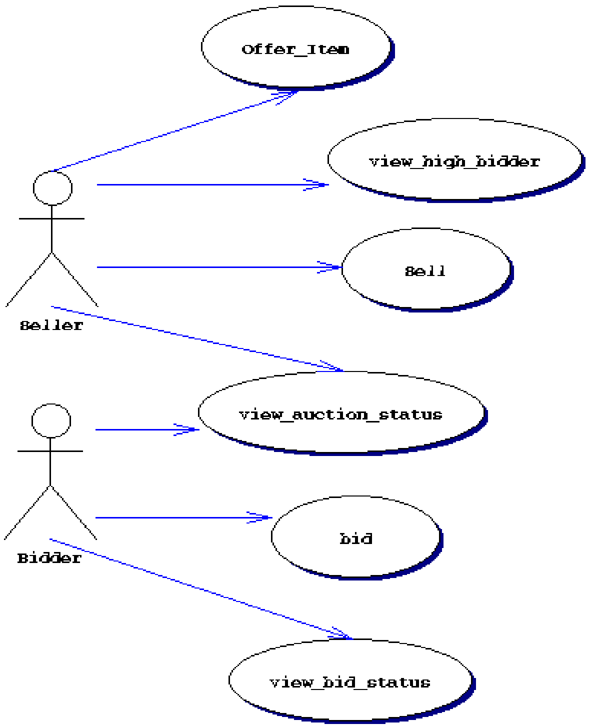

Fig. 1. Case Scenarios for Seller and Bidder's Functionality in Phase I

- **Offer-Item**: If the auction is currently empty, any user may offer an item for sale by providing a user ID, an item description, and an optional initial price. The initial price becomes the current price. This operation is not permitted if the auction is currently busy.
- **Sell**: If at least one Bid has been accepted, the Seller may sell the item at the current price. The actual sales transaction is assumed to take place outside the scope of the Auction. The Seller must identify itself by providing the same userID that was used in the offer-item operation.
- **Bid**: Any user, other than the seller, may place a bid on the current item by supplying a user ID and a bid price. If the price bid is higher than the current price, the bidder becomes the current high bidder, and the price becomes the current price.
- **view-high-bidder**: The identity of the current high bidder, if any, is viewable only to the current Seller.
- **view-bid-status**: A bidder can determine at any time whether it is the current high bidder. This operation will fail if the auction is not currently active (no item is currently for sale, or the current item is already sold).
- **view-auction-status**: Any user may view the status of the auction. Status includes auction state (empty/active), and if the auction is active, the description of current item and current price. The identity of the current high bidder is viewable only to the seller.

IV. Phase II - Auction Server with Multiple Simultaneous Auctions

The purpose of this phase is to extend our working knowledge of CORBA by implementing a factory/finder that generates new auctions and returns remote references to them. *We'll also deploy the user interface as an applet*. This eliminates the need to deploy client code to client machines, at the cost of some configuration complications and having to make sure you are using browsers that support the correct Java version.

Our goal is to extend the auction application we started in phase I to allow multiple simultaneous auctions, and to provide each user with an integrated seller's and bidder's interface. In addition, we want to make the clients a little more dynamic; when a bidder's bid is topped by another bidder, or when the seller sells the item, we want the bidder to be notified in a timely manner. Using VisiBroker and its Gatekeeper, you can accomplish this with a callback interface. Note that callbacks to applets are not supported by the current version of IIOP and the Java applet environment, so it may be necessary to implement a polling solution, if not using Visibroker. The next version of IIOP is expected to solve this problem.

General Requirements, beyond those of Phase I:

- Multiple active auctions must be supported, each with one seller and potentially many bidders.

- A given user may be playing seller and bidder roles in separate auctions.
- The client must be an *applet* that supports both bidder and seller interfaces. It is optional whether this is done with a single applet or multiple applets in the same page.
- Changes in auction status must be delivered to the user interface in a reasonably timely fashion without the need for user initiative.

Detailed requirements for this assignment are outlined by the Use Cases below:

- **Auction-status**: Each registered user will be presented with two lists of current auctions.
  - Those auctions in which the user is in the Seller role – the auctions that were initiated by that user.
  - All other auctions on which the user could potentially bid.

  For each auction, the user is able to view the item description and the current price. The user will be able to manually update the display at any time, retrieving current data on all auctions. For auctions in which the user is in the Seller role, the current price will be automatically updated whenever a new bid is submitted. For auctions in which the user is in the Bidder role and has the current high bid, the Bidder will be automatically notified if the bid is accepted by the Seller, or if the bid is superceded by a higher bid. Once an auction is closed (the item has been sold), it shall be removed from the displays of all users. It is acceptable if this removal is deferred until the next explicit user-initiated refresh operation.
- **Bid**: Any user may play the Bidder role with respect to any auction that was initiated by some other user in the Seller role. As long as the item has not yet been sold, the Bidder may place a new bid. That bid will be accepted if it is the current highest bid at the time it is processed by the auction.
- **Registration**: Each user of the system is identified by a login name and a password. Once a given login name has been associated with a password, that same login name may not be used with a different password.

  Login names are required to be unique; they will be the identifiers that associate Sellers and Bidders with Auctions.

  Users may be either logged in or logged out. When logged in, their displays will be automatically updated as described in the Auction-status use case. Each user may be logged at most once at any given time. In other words, at most one display needs to be updated for any given user.

  The following operations shall be supported:
  - *Log in new user*: If the user name is unique, then the user will be prompted to supply a password.
  - *Log in existing user*: If the user name is not unique, then the password must match the one given when that user name was first given as a new user.
  - *Log user out*: A user may depart from the system at any time. No information is lost from the server when this happens.
  - *Automatic user logout*: It must be possible to determine whether a user has shut down the client without an explicit logout. This is because remote clients can become disconnected in ways that cannot be managed at the client end.
- **Sell-items**: A user becomes a Seller by creating a new auction. Required information for a new auction includes the seller's ID, the description of the item being sold, and the initial price of the item.

  The seller is kept informed of the current high bid and the identity of the high bidder for each auction in which it is the seller. Users other than the seller are not permitted to view the identity of the high bidder for any auction. The only exception to this is that any user may determine whether it is the current high bidder in any auction.

  After at least one Bidder has placed a bid on the item in an Auction, the seller may sell the item to the current high bidder. This will be the high bidder at the time the auction processes the sell operation, and not necessarily the high bidder that is currently displayed to the seller.
- **Actor Bidder**: Any User may play the role of Bidder in any Auction for which they are not the Seller.
- **Actor Seller**: Any User may initiate a new Auction, thereby becoming the Seller for that Auction.
- **Actor User**: Users must "register" or log in before they can use any other portions of the system. In other words, a user who has not logged in cannot invoke any functionality other than the login operation.

V. Phase III - Auction Server with Support for Financial Transactions

In this phase of the project we will extend the multiple-auction server of Phase II with a Bank server. Banks have accounts, and accounts have balances. When you bid in an auction, you must have enough money in your account to cover the bid. If someone tops your bid, you then have additional uncommitted money available to support more bids. When an item is sold, money is transferred from the buyer's account to the seller's account automatically. Deposits may be made at any time, while withdrawals must not exceed the uncommitted balance.

General Requirements beyond those of Phase II:

- Separate bank and auction services will be required. It must be possible to run them on different platforms in separate file systems.
- Users must establish accounts in the Bank before registering for the Auction.

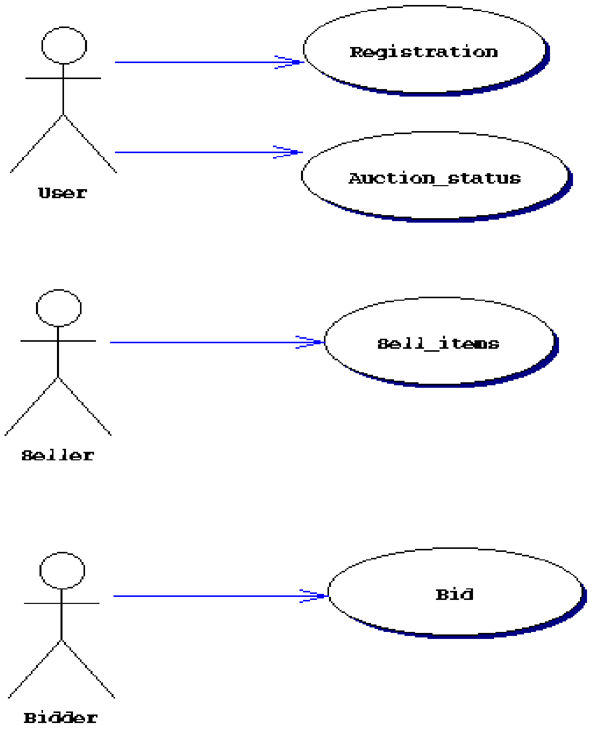

Fig. 2. Case Scenarios for Seller and Bidder's Functionality in Phase II

Detailed requirements for this assignment are outlined by the Use Cases below.
Analysis Types:

Bank	Set of Accounts
Account	User Identity (consisting of Username and Password), Current Balance ≥ 0, Committed Balance $\leq$ Current Balance
Auction Manager	Set of Auctions
Auction User	Username (length ≥ 3 chars), Password (length ≥ 6 characters), Bank Account (Account)
Auction	Seller (Auction User), Item Description, Current Price, Current High Bidder (Auction User)

Use Cases: Bank
All user access to accounts requires username/password verification.

- Create account: The user provides a name and password to the bank and receives a newly-created account.
  - Only one account may be assigned to a particular username. The Bank will refuse to create an account if an account already exists under the given username, even if the given password differs from the one assigned to the existing account.
  - The account is created with current balance = 0 and committed balance = 0.
  - The username and password are saved with the account.
- Manage account: The user or the auction may check the balance, deposit money, or withdraw money from the user's account. User access to the account is controlled by matching name and password. Balance information includes both current balance and committed balance. Withdrawals must not result in current balance less than committed balance.
- Update commitment: The auction may increase or decrease the committed balance. It is not permitted to allow the committed balance greater than the current balance.
- Close account: The user may close out its account and receive the remaining current balance just in case the committed balance is 0.

Use Cases: Auction

- **Register**: Each user of the system is identified by a login name and a password. Once a given login name has been associated with a password, that same login name may not be used with a different password.

  Login names are required to be unique; they will be the identifiers that associate Sellers and Bidders with Auctions.

  Users may be either logged in or logged out. When logged in, their displays will be automatically updated as described in the Auction-status use case. Each user may be logged at most once at any given time. In other words, at most one display needs to be updated for any given user.

  Each user must have an associated Bank Account. There are two options for associating an account with a user in the auction:

  - The Auction may request the user's bank username and password, which may be used to access the account.
  - The bank may provide a function to retrieve an opaque reference to a user's account, which may then be passed to the auction.

  The following operations shall be supported:

  - Log in new user: If the user name is unique, then the user will be prompted to supply a password and a bank account.
  - Log in existing user: If the user name is not unique, then the password must match the one given when that user name was first given as a new user.
  - Log user out: A user may depart from the system at any time. No information is lost from the server when this happens.
  - Automatic user logout: It must be possible to determine whether a user has shut down the client without an explicit logout. This is because remote clients can become disconnected in ways that cannot be managed at the client end.

- **View status**: Each registered user will be presented with two lists of current auctions.

  - The first list is those auctions in which the user is in the Seller role – the auctions that were initiated by that user.
  - The second list is all other auctions on which the user could potentially bid.

  For each auction, the user is able to view the item description and the current price. The user will be able to manually update the display at any time, retrieving current data on all auctions. For auctions in which the user is in the Seller role, the current price will be automatically updated whenever a new bid is submitted. For auctions in which the user is in the Bidder role and has the current high bid, the Bidder will be automatically notified if the bid is accepted by the Seller, or if the bid is superceded by a higher bid. Once an auction is closed (the item has been sold), it shall be removed from the displays of all users. It is acceptable if this removal is deferred until the next explicit user-initiated refresh operation.

- **Sell item**: A user becomes a Seller by creating a new auction. Required information for a new auction includes the seller's ID, the description of the item being sold, and the initial price of the item.

  The seller is kept informed of the current high bid and the identity of the high bidder for each auction in which it is the seller. Users other than the seller are not permitted to view the identity of the high bidder for any auction. The only exception to this is that any user may determine whether it is the current high bidder in any auction.

  After at least one Bidder has placed a bid on the item in an Auction, the seller may sell the item to the current high bidder. This will be the high bidder at the time the auction processes the sell operation, and not necessarily the high bidder that is currently displayed to the seller. At the time of sale, the following operations are performed within a single transaction:

  - The status of the auction is changed to Sold.
  - The amount of the high bid is transferred from the bidder's bank account to the seller's bank account, updating both balances.
  - The committed balance of the bidder's bank account is reduced by the amount of the sale.

  After the sale has been successfully completed, the seller may inspect the auction to determine the identity of the buyer and the amount of the sale. It is unacceptable for the seller to assume that the identity of the buyer and the sale amount are unchanged from the information available at the time the sell operation is initiated.

- **Bid**: Any user may play the Bidder role with respect to any auction that was initiated by some other user in the Seller role. As long as the item has not yet been sold, the Bidder may place a new bid. That bid will be accepted if it is the current high bid at the time it is processed by the auction.

  When a bid is accepted, three operations are performed within a single transaction:

  - The identity of the current high bidder and the amount of the current high bid is updated in the auction.
  - The Bidder's current commitment is increased by the amount of the bid. If the resulting current commitment is higher than the current balance in the account, the operation is rejected and the bid is not accepted.
  - If the bid is accepted and there is an existing high bidder, the current commitment of the existing high bidder is reduced by the amount of the existing high bid.

VI. What to Hand In

A. Design

Before you start hacking away, plot down a design document. The result should be a system level design document, which you hand in along with the source code. Do not get carried away with it, but make sure it convinces the reader that you know how to attack the problem.

B. Hard Copy of Portions of Source code

Hand in a hard copy of all the code you created. Try to condense the printouts, e.g. with `enscript -2r <file>` The code should be easy to read (read: well-commented!). The grader reserves the right to deduct points for code that he/she considers undecipherable.

C. Measurements

In order to compare the efficiency of your implementation, you will perform some measurements. I will leave it to you to decide what measurements should be done.