

Problem 3: Distributed Auction Service

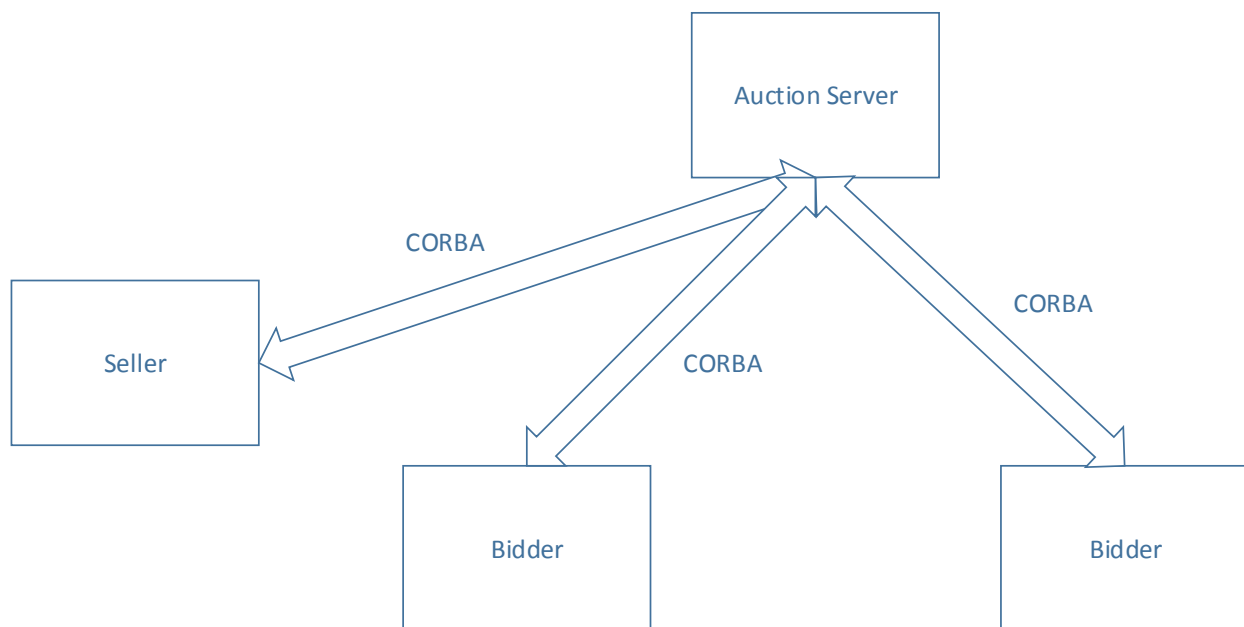
Chris Baumler

Description:

The distributed auction service is designed to allow buying and selling of items using an English auction protocol. The system consists of a distributed auction server that allows one item to be auctioned at a time and several clients. Two types of clients are supported - bidders and sellers. Sellers are able to post items to be auctioned, and bidders are able to place bids on the items.

Auction Server: The auction server manages the current auction. It keeps track of the highest bidder and updates the price of the auction item.

Client: The client provides two different interfaces. If no item is currently being auctioned, a client may offer an item to be sold. This client becomes the seller and may choose to sell the item at any time provided at least one bid has been placed. If an item is currently being auctioned, any client that is not the seller may place bids on the item.



Requirements:

Server:

If no auction is currently active, the auction server **shall** allow a single client to place an item up for sale.

If a client places an item up for sale, the auction server **shall** activate an auction to sell the item.

If a client attempts to place an item up for sale, but an auction is already active, the auction server **shall** return an error to the client.

If an auction is currently active, the auction server **shall** accept bids from clients other than the seller.

If a client attempts to bid when no auction is active, the auction server **shall** return an error.

If a seller attempts to place a bid on an item, the auction server **shall** return an error.

When a new high bid is placed by a client, the auction server **shall** update the price of the item for sale.

The auction server **shall** keep track of the highest bidder.

If a client requests to view the highest bidder when no auction is active, the auction server **shall** return an error.

If a client other than the seller requests to view the highest bidder, the auction server **shall** inform the client whether it is the highest bidder.

If the seller requests to view the highest bidder, the auction server **shall** return the highest bidder.

If the seller chooses to sell the item currently being auctioned, and at least one bid has been received, the auction server **shall** end the auction.

If the seller tries to sell the item currently being auctioned, and no bids have been received, the auction server **shall** return an error.

If a client other than the seller requests to sell an item, the auction server **shall** return an error.

If a client requests to sell an item when no auction is active, the auction server **shall** return an error.

If the seller requests to view the auction status, the auction server **shall** respond with a status containing the auction state (empty/active), and if the auction is active, the description of the current item, the current price, and the current high bidder.

If a client other than the seller requests to view the auction status, the auction server **shall** respond with a status containing the auction state (empty/active), and if the action is active, the description of the current item and the current price.

Client:

The client **shall** provide an interface for the user to place an item up for sale, specifying a user ID, an item description, and an optional initial price.

The client **shall** provide an interface to view the highest bidder for the current auction.

The client **shall** provide an interface to sell an item.

The client **shall** provide an interface to view the status of the current auction.

The client **shall** provide an interface to place a bid.

Instructions for Use:

The following instructions assume the server IP address is 192.168.222.135. Please substitute the appropriate IP address.

Launching the Corba Daemon:

```
orbd -ORBInitialHost 192.168.222.135 -ORBServerHost 192.168.222.135 -  
ORBInitialPort 1050&
```

Launching the Auction Manager Daemon:

```
java -jar AuctionManager.jar -ORBInitialHost 192.168.222.135 -  
ORBServerHost 192.168.222.135 -ORBInitialPort 1050
```

Launching the client applet:

Click on AuctionClient.html.

Note: AuctionClient.jar, AuctionCorbaLayer.jar, and forms-1.3.0.jar must be in the same directory as the html file.