2009 International Conference on Computer Engineering and Applications
IPCSIT vol.2 (2011) © (2011) IACSIT Press, Singapore

Online Auction Software Fundamentals

Jarrod Trevathan[+], Wayne Read and Rodel Balingit

Discipline of Information Technology, School of Business, James Cook University

Abstract. Online auction technologies are revolutionizing the way we conduct business online. However, the proliferation of auction related research and the development of new auction security techniques are often limited by the uncooperative behaviour of the major online auctioneers. Due to the limited availability of quality auction data and literature on online auction software design, we have created our own online auction server to facilitate auction research. This paper describes our experiences with designing online auction software. We present a basic online model upon which more sophisticated auction scenarios can be crafted. We address website design, preliminary security, the database schema, transaction and timing issues, and also show how software bidding agents can interact with the model. Our model is grounded in object oriented techniques and is open source so that other researchers can expand upon our approach.

Keywords: e-commerce, Crow's Foot, software bidding agents, timing

1. Introduction

Online auctioning is unparalleled as the fastest growing exchange medium to emerge from e-commerce technology. The English auction is the most renowned type of auction and coincidentally the most popular form of auction conducted online. However, despite its popularity there is limited available literature on software design for online English auctions except for Wellman and Wurman [2] and Kumar and Feldman [3]. This is largely due to the dominance of commercial online auctioneers such as eBay[1] and Yahoo Auction[2]. Commercial auctioneers tend not to publish their research nor are they willing to share data.

We required our own infrastructure for performing research. There are many vendors selling auction software online. However, such software is expensive and can not be customised for our research needs. Due to this and the limited availability of academic literature on auction software design, we have created our own online auction server. This paper describes our experiences with designing online auction software and present a client-server software model for conducting standard online English auctions. The model is based on an existing auction server developed at James Cook University called the "Research Auction Server" (RAS) [4]. RAS is used to perform both simulated and real auctions to gather data on the performance of key areas of auction research. RAS has been used to conduct research in auction security, fraudulent bidding behaviour, bidding agent design and market clearing algorithms. RAS is available online at http://auction.maths.jcu.edu.au.

This paper is organised as follows: Section 2 describes the mechanics of a typical auction, the major software components of an online auction, and the processes involved. Section 3 discusses website navigation. Section 4 presents an object model of the system. Section 5 describes the database schema, transaction issues and timing concerns. Section 6 outlines an interface for software bidding agents. Section 7 provides some concluding remarks.

[+] Corresponding author. Tel.: + 61 07 4781 6905; fax: +61 07 4781 4028.
E-mail address: jarrod.trevathan@jcu.edu.au.

[1] http://www.ebay.com

[2] http://www.yahoo.com/auction

2. Online Auctions

There are many types of auctions such as English, Vickrey, Dutch, double, etc. (see [1], [4]). The English auction is the most well known type of auction. Formally an English auction is referred to as an ascending price auction. To win, bidders must outbid each other forcing the price up. The winner is the bidder with the highest price at the end of the auction.

The form of a bid in an English auction is: < *bidder, price, time* >. At any time in an online auction, a bidder can request a price quote from the Auctioneer. The *price quote* contains the bid information of the current highest bid. An auction's *bid history* is a list of all bids that have been submitted. The following table presents an example bid history for an auction. Here the winning bid is $ 45.

Bid	Bidder	Price	Time
7	Darna	$ 45	33:05
6	Jerry	$ 40	30:15
5	Tess	$ 30	24:22
4	Darna	$ 25	17:01
3	Tammy	$ 15	12:35
2	Mr Gab	$ 10	05:44
1	Jerry	$ 5	02:09

English auctions can have three additional parameters:

- *Starting price* - the minimum price at which the bidding must commence;
- *Reserve price* - the minimum price the seller of the item will accept. If the final price is below the reserve, the auction result is void; and
- Minimum *bid increment* - the minimal amount required to outbid the current highest offer.

Online auctions allow many variations on English auctions. For example, bidders can alter/cancel bids, auction multiple goods, bidders can issue sell bids, etc. Previous auction software architectures have strived to implement as many auction types as possible. However, the model in this paper only focuses on English auctions. There are several reasons for this. First of all, English auctions are the most common type of auction performed online. Secondly, there is limited literature on software design for online English auctions. Furthermore, some auctions (e.g., continuous double auctions) are more complicated than English auctions. This burdens the reader with extraneous detail regarding auctioning mechanisms. Therefore we do not aspire for our model to be generic at the expense of having to include excessive detail.

2.1. Components of an Online Auction

Figure 1 presents a high level software model for performing online auctions. There are two main parties: a bidder and an Auctioneer. The parties are joined by a communication link.

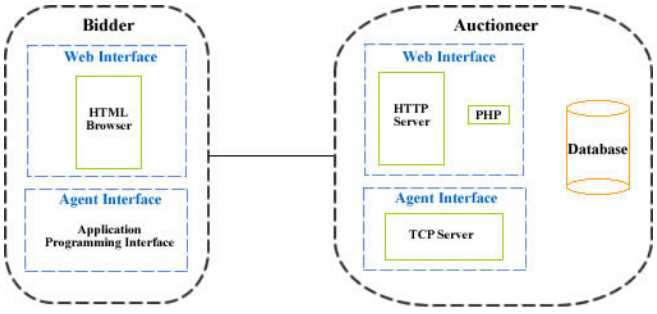

Fig: 1 Online Auction Software Model

There are two types of interface for a bidder. The first is the web interface. This is for a human bidder. The bidder interacts with the Auctioneer via a HTML browser. The second interface is for a software bidding agent. A bidding agent interacts with the Auctioneer using an application programming interface.

The Auctioneer runs a web server (e.g., Apache) and a scripting language, in this case php. The entire auction is database driven. All state information (e.g., bids, timing, etc.) about the auction is contained in the database. When a client submits a bid or requests a price quote, a database transaction occurs. The database generates dynamic web pages in response to bidder activity using the scripting language.

2.2. Processes Involved

There are several main activities in an online English auction:

- *Initialization*: The Auctioneer sets up the auction and advertises it;
- *Registration*: In order to participate in the auction, bidders must first register with the Auctioneer;
- *Price quote*: A bidder obtains a price quote from the Auctioneer;
- *Bidding*: A bidder submits a bid to the Auctioneer;
- *Winner Determination*: The Auctioneer determines the winner according to the auction rules;
- *Transaction settlement/payment* is the process of collecting payment from and delivering the goods to the winning bidder; and
- *Bidder/Winner Notification* refers to the processes involved with informing a bidder with information other than the price quote. This is information specific to an individual such as confirmation of bid receipt or notifying the winner that they have won.

3. Web Interface Navigation

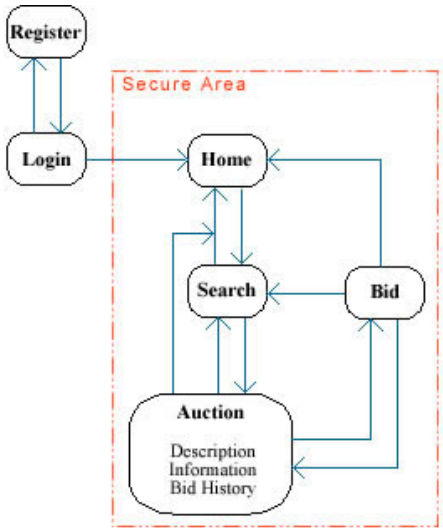

Fig: 2 Web Interface Navigation

Figure 2 illustrates how human bidders navigate the auction website. Each bubble represents a major section of the site. Lines represent links between sections and arrows indicate the navigational direction a user must follow to reach a particular page.

After registering, a user can log in using a password. Upon login a user enters the secure area (shown by the dash box in Figure 2). This is a collection of pages which contains operations that only a registered user can perform.

The home page is the main area for the bidder. Bidders are able to search for a listed auction. Once a bidder has selected an auction, they are able to obtain a description of the item and information regarding the auction (i.e., price quote, minimum increment and time remaining). The bidder can then use this information to submit a new bid. After submitting a bid, a bidder can return to the home page, search for a new auction or return to the current auction.

In the secure area specific information regarding the user (such as user id and password) must be carried from one page to another. To achieve this, the secure area is implemented using session variables. Session variables are similar to cookies, as they are used to store information for a particular period of time. The values in session variables exist until the session terminates. This allows information to be passed between web pages or to another web site. A session can be created using a session identifier which is stored on the server. When the client makes a request, the data stored in the session variable can be accessed any number of times until the session ends.

An alternative approach for passing information between pages is to use a query string. However, using a query string in this manner is difficult and cumbersome. Session variables on the other hand allow information to be more easily retrieved and maintained. When a bidder logs off the session variable is destroyed. The variable must also be destroyed after a predetermined amount of bidder inactivity (for example, when a user forgets to logoff).

Upon submitting a bid, the bidder receives an email confirming that his/her order has been received. The bidder will also be informed of such information via the web page.

4. Object Model

The web interface for RAS allows human bidders to interact with the Auctioneer. Figure 3 depicts an object model for RAS's web interface. There are two objects: Bidder and Auctioneer. The top portion of each object indicates its internal state and the bottom portion lists the methods that it can use.

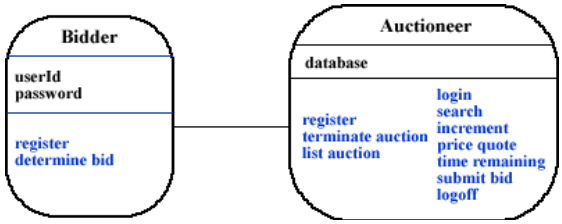

Fig: 3 Web Interface Object Model

A *Bidder's* internal state is his/her userId and password. The bidder interacts with the Auctioneer via the Auctioneer's methods. The bidder only has two methods *register* and *determine bid*. During registration, a bidder obtains a userId and password which they keep secret. When participating in an auction a human bidder then determines his/her bid according to individual preferences.

The *Auctioneer* manages the database which is the central component of the auction. This contains all information regarding bidders, their bids and auctions conducted. There are two types of methods an Auctioneer can perform: private and public. Private methods are operations that only the Auctioneer can perform such as listing a new auction (*list auction*), *register* new bidders and *terminate* an auction. Public methods are services which are requested by a bidder. This includes the ability to *login*, *search* for an auction, request the minimum *increment*, obtain a *price quote*, request the *time remaining*, submit a bid (*submit bid*) and *logoff*.

5. Database

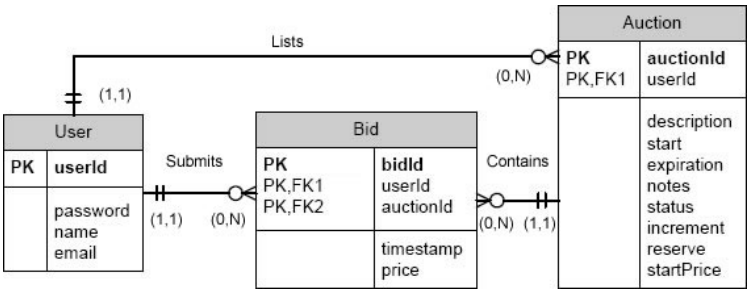

Fig: 4 Online Auction Database Entity Relationship Diagram

Figure 4 presents an Entity Relationship Diagram (ERD) for the database in an online auction. There are three entities: *user*, *auction*, and *bid*. The primary key for each entity is underlined.

When a bidder registers, their details are entered into the user entity. Each user is identified by a unique *userId*. Information stored about a user includes their *password*, *name* and *email* address.

Each auction is identified by a unique *auctionId* and is listed by seller that is identified by his/her *userId*. Information stored about an auction includes an item *description*, *start* time, *expiration* time, *startPrice*, *reserve* price, minimal *increment* and *notes* about the item. The *status* attribute indicates whether the auction is currently active or has terminated.

Every time a bidder submits a bid, all information about the bid is entered into the bid entity. Each bid is identified by a unique *bidId*. To associate a bid with a bidder, the bid entity stores the *userId* as a foreign key. To associate a bid with an auction, the bid entity also stores the *auctionId* as a foreign key. Other bid information includes the *price* of a bid and a *timestamp* that indicates when the bid was submitted.

There are several database transactions which occur during an auction:

- Initialization - A new auction's details are entered into the database.
- Registration - A new user's details are entered into the database.
- Login - An existing user's details are retrieved from the database.
- Price Quote - An existing auction's details and bid history are retrieved from the database.
- Bid Submission - A new bid's details are entered into the database.
- Auction Termination/Winner Determination - An existing auction's details and bid history are retrieved from the database. The auction's status is set to 'inactive' and entered in the database. The winner is determined according to the auction's rules.

5.1. Timing

When a bid is received for a particular auction, it is compared to the current highest bid. If it does not meet the minimal required amount to outbid the existing highest bid, then it is discarded. A bid that does meet the required amount is timestamped by the Auctioneer and entered into the database.

Online English auctions can terminate according to the following rules:

- Expiration Time - The auction closes at a predetermined expiration time;
- Timeout - The auction closes when no bids higher than the current highest bid are made within a predetermined timeout interval; or

- Combination of Expiration and Timeout - The auction closes when there is a timeout after the expiration time.

In our model, the auction is database driven. Rather than having a running program continually check the time in order to terminate the auction, our auction terminates in response to bidder activity. This means that every time a bidder submits a bid or requests a price quote a script is run to check whether the auction has terminated according to the database clock. This involves comparing the database time to the expiration attribute of the auction entity in the auction ERD (Figure 4). A pseudocode script for terminating an auction with an expiration time is shown below:

```
if database time >= expiration {
    set auction status to inactive
    select highest bid from auction
    if bid > reserve
        declare this bid the winner
    else
        auction is void
}
else {
    expiration = expiration + timeout
}
```

This setup does not require synchronization between the bidders and the Auctioneer.

Terminating an auction using a timeout involves adding additional time to the auction's expiration attribute in the database every time a higher bid is received. Let ***timeout*** be a constant time increment. Adding a timeout extends the terminate auction script as shown by the red section of the code.

Some variants of English auctions allow *bid cancellation*. This is common in English auctions which terminate using an expiration time. The justification being that such auctions can often take days or weeks. In this situation, a bidder may be reluctant to make such an open-ended bid.

The simplest approach to bid cancellation is to delete the cancelled bid from the database. However, for security reasons a cancelled bid should be recorded. This is achieved by including *valid* and *cancellationTime* attributes to the bid entity of the ERD shown in Figure 4. Figure 5 illustrates new additions to the bid entity where the red indicates the new attributes.

A bid is initially in a valid state and the cancellation time is *null*. This indicates that it is to be considered for an auction. When a bid is cancelled, it enters an invalid state indicating that it should not be considered for an auction. This allows the bid details and time of cancellation to be recorded but not included in an auction.

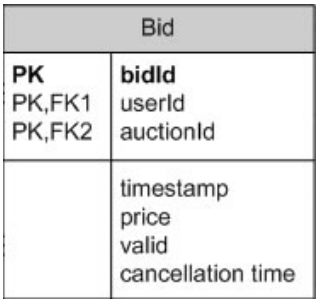

Fig: 5 Bid Entity with Bid Cancellation

6. Bidding Agents

Software bidding agents bid on behalf of a human bidder according to a predetermined strategy. In terms of an English auction, a bidding agent is permitted to outbid any bid until the bidding price exceeds a maximum amount specified by the human bidder. In auctions that can last days or weeks, bidding agents remove the need for a bidder to constantly observe an auction. Instead a bidding agent monitors the auction proceedings for any price activity and responds in accordance with its programmed strategy.

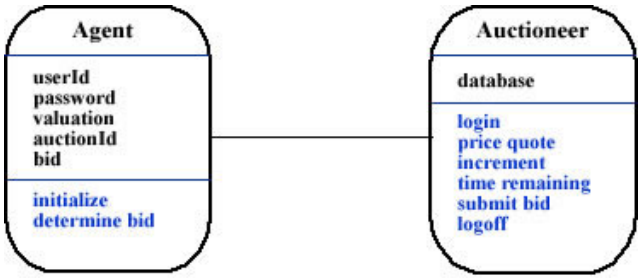

Fig: 6 Software Bidding Agent Object Model

Figure 6 presents the object model for a simple bidding agent. The Auctioneer object must provide an agent with six basic services: *login*, request *price quote*, minimum *increment*, *time remaining*, *submit bid* and *logoff*. We have not listed the Auctioneer's private methods in the model.

Similar to a human bidder, an Agent object has a userId and password. An agent also knows the auctionId of the auction it wants to participate in. The valuation is the maximum limit that an agent can bid. An agent can perform two operations: *initialize* and *determine bid*. The initialize method provides the agent with the human's userId and password, and instructs the agent which auction to participate in and the valuation of the auctioned goods. The following pseudocode script of the *determine bid* method illustrates the basic operation of a bidding agent.

```
bid = 0
 login ( userId, password, auctionId )
repeat {
     if ( bid <= price quote ) {
          bid = price quote + increment
          if ( bid <= valuation AND time remaining )
               submit bid ( bid )
          else
               logoff
     }
 }
```

Initially the agent's bid is zero. The agent logs in by supplying its userId and password. The agent also indicates the auction it wants to participate in by specifying the auctionId. Once logged in, an agent repeatedly requests a price quote and alters its strategy accordingly.

In our simple agent example, the agent's current bid is compared to the current highest bid in the auction. If the agent's bid is less than the price quote, then the agent increases its bid by a given increment. If the new bid does not exceed the agent's preset valuation, then the agent submits the bid to the Auctioneer. Otherwise the agent logs off as it is not allowed to exceed the valuation. Before submitting a bid, the agent also checks whether the auction has terminated. If so, the agent logs off.

Bidding agents are not limited to the strategy employed by our simple agent example. Bidding agents can employ sophisticated strategies. Some of these strategies are based on statistical analysis of the auction data and neural network reasoning about the best strategy to take.

7. Conclusions

Despite the popularity of auctioning online, there is limited documentation regarding online auction software design. This paper discussed our experiences with creating online auction software. We presented a software model for conducting online English auctions. The model is based on an existing auction server called RAS which we use to gather auction data for our research.

As our auction model is specific to online English auctions, it is simple and concise compared to previous literature on auction design. Our system is modelled using an object-oriented approach. We described the main auctioning processes, web navigation, database, timing and bidding agent issues for an online English auction. RAS is used to test the efficiency and practicality of security and anonymity mechanisms in online auctions. RAS has been used to perform hundreds of simulated and real auctions.

8. References

[1] M. Franklin and M. Reiter. The Design and Implementation of a Secure Auction Service, *IEEE Transactions on Software Engineering*, vol. 22, Pages 302-312, May 1996.

[2] M. Kumar and S. I. Feldman. Internet Auctions, in *Proceedings of the 3rd USENIX Workshop on Electronic Commerce*, Pages 49-60, August 1998.

[3] M. P. Wellman, P. R. Wurman and William E. Walsh. The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents, *Second International Conference on Autonomous Agents (AGENTS)*, Pages 301-308, May 1998.

[4] J. Trevathan and W. Read. – Shilled – Fraud and Security in Online Auctions, VDM Verlag, Germany, ISBN 9783639054712, 168 pages, 2008.