In *First IEEE Workshop on Dependable and Real-Time E-Commerce Systems (DARE-98),*
Denver, CO, USA, June 1998

Real Time Issues for Internet Auctions

Michael P. Wellman **Peter R. Wurman**
University of Michigan
Artificial Intelligence Laboratory
1101 Beal Av, Ann Arbor, MI 48109-2110 USA
{ wellman, pwurman } @umich.edu

Abstract

Designers of online auction mechanisms face many interesting choices related to the timing of events. We discuss several temporal issues that arise in the context of auction mechanisms, and argue that it is generally advantageous to design for maximal asynchrony and robustness to network delays. The design of a configurable auction server (the Michigan Internet AuctionBot) exemplifies these principles, and our experience in operating the AuctionBot illustrates the consequences of this approach for real-time performance.

1 Introduction

In the past two years, Internet auctions have gone from nothing to an activity measured in hundreds of millions of dollars. As of March 1998, Yahoo (`www.yahoo.com`) lists over 90 online auctions, and this list is far from complete. Compared to projections of various forms of electronic commerce, auctions have been among the most successful medium. Indeed, by any measure, a far greater fraction of exchange happens via auctions online than it does offline.

Speculations abound regarding the source of the popularity of online auctions. For some, it is a marketing gimmick—enticing customers by making a game of the buying process. Indeed, participating in auctions can be fun, and this factor undoubtedly plays a significant role. More fundamentally, however, auctions support dynamic formation of prices, thereby enabling exchanges in situations where a fixed price—unless it happened to be set exactly right—would not support as many deals. This is, of course, the rationale for auctions in offline contexts as well. The online environment is particularly conducive to auctions, due to at least two important properties of the electronic medium.

First, the network supports inexpensive, wide-area, dynamic communication. Although the primitive communication protocol is point-to-point, a mediating server (i.e., the auction) can easily manage a protocol involving thousands of participants. Moreover, the information revelation process can be carefully controlled. Unlike the human auctioneer orchestrating a room of shouting traders, a network auction mediator can dictate exactly which participants receive which information and when, according to auction rules.

Second, to the extent that auction-mediated negotiation is tedious, it can be automated. Not only the auctioneer, but also the participating traders, may be represented by computational processes.

Although this second opportunity has yet to be exploited by popular Internet auctions, we believe that ultimately online auctions will be primarily the province of programmed traders. Our auction server—the Michigan Internet AuctionBot [10]—was built in large part to investigate the potential of computational markets to mediate the interactions of software agents.

2 Auction Types

Say “auction” to a layperson and you will likely conjure one of several images. Picture the hushed room of well-dressed art buyers attending to a distinguished individual standing at the podium with a gavel, saying “going once...”. Or maybe an outdoor venue presided by a slick auctioneer, speaking with unintelligible rapidity, pointing to the livestock as they come up for sale. Less ingrained, but also commonplace, is the auctioneer at the fishing dock lowering the price until somebody agrees to haul away that day’s catch. These are all instances of *open outcry* auctions, defined by the mode of communication in which all status information is conveyed immediately and globally to all participants.

Another form of open outcry auction is the familiar “trading pit” of a commodities or securities exchange. Although this might not be viewed as an auction in common parlance, it clearly falls well within the scope of the technical definition of auction [2]—essentially, any well-specified mechanism for determinating the terms of an exchange.[1] Even the seemingly chaotic trading pit operates according to rules governing who is allowed to shout what and when, and what the shouts entail in terms of offers of exchange. The distinguishing feature of the trading pit is that it is *two-sided*: both buyers and sellers play bidders in this protocol. In contrast, the art, livestock, and fish auctions alluded to above are *one-sided*: a single seller offers an item to multiple bidding buyers.

Unlike open outcry events, in *sealed-bid* auctions the participants do not learn the status of the auction until the end (if then), or until some other explicit action by the auctioneer. Familiar examples of sealed-bid auctions include government sales of leases for offshore oil drilling, or procedures by which real-estate developers let construction contracts. In the latter example, note that it is the sellers (of construction services) doing the bidding. Sealed-bid auctions may be one-shot, or may involve complex iterations over multiple rounds, as in the prominent US FCC spectrum auctions held in recent years [3].

Like open outcry, sealed-bid auctions may also come in one-sided or two-sided varieties. Both classes of auctions may also be further distinguished according to their particular rules of operation, specifying such features as bidding restrictions, event timing, information revelation, and pricing and allocation policies. Indeed, all of the commonly studied auction types [1, 2, 4] can be placed within a taxonomy of possible auctions. We based our configurable auction server on a particular parametrization of this space of possibilities [5, 10]. A more specific parametrization of the class of descending (i.e., decreasing price) open outcry auctions defines the FM96.5 ("FishMarket") auction testbed [6].

3 Internet Auctions

3.1 Real-Time Open Outcry

Like their offline counterparts, Internet auctions come in a range of types and variations. Almost all are one-sided (i.e., allow buy bids only), run by the sellers themselves or by third parties.[2] Most attempt to mimic the familiar open outcry types, for example by posting the current high bid and bidder's identity. This very familiarity is a sufficient reason to adopt this approach.[3]

However, few (if any) popular sites attempt to conduct protocols that are "real-time" open outcry in the sense that offline auctions tend to be. To do so would require identifying the group of participants, and broadcasting all bid messages (perhaps using multicast communication protocols, or so-called "push" technology,[4] for example). Although somewhat more complicated, the main impediment to this scheme is not technological, but a matter of flexibility. In the offline world, holding an open outcry auction means getting all the participants together in the same place and time. Using the Internet (or other communication technology, for that matter) can avoid the need to collocate, of course. Flexibility in temporal coordination is equally important. Internet users would far prefer to attend to auctions at their own convenience, rather than tune in at a designated uniform time.

In the early days of Internet auctions we observed a few attempts to hold such real-time auction events. None of these efforts have succeeded or persisted, to our knowledge. (However, recently we have seen some elements of this approach reappear in auction prototypes, enabled by extensions in world-wide web protocols.)

Note that for auctions designed to be used by software agents, the temporal flexibility issue for participants may be ameliorated somewhat. That is, it may be more plausible to expect software agents to monitor an auction channel during a particular interval than it would be for human agents. Nevertheless, the synchronization required to accomplish this is still costly, and it is unclear what advantage generally accrues. Thus, we do not expect that online auctions directly replicating the real-time character of offline open outcry auctions will become prevalent, even for software agents.

3.2 Sealed-Bid Mechanisms

If bidding participants attend to the auction asynchronously, the distinction between outcry and sealed-bid auctions becomes somewhat blurred. We can view the auction process in at least two ways:

1. As an outcry auction, where the revision in price

[1] Technically, the auction *mechanism* is that part of the *protocol* governing the behavior of the auctioneer, essentially defining the rules of interaction, and determination of the result. The entire auction protocol comprises the mechanism, plus the behaviors of participating bidders.

[2] See Ungar et al. for a recent discussion [7]. We have argued that introducing two-sided auctions can add significant flexibility without much additional complexity [9].

[3] Indeed, our experience with the AuctionBot suggests that most Internet users have little patience with complex or unfamiliar auction rules. Most Internet auctions' documentation pages spend more space on shipping and liability issues than on bidding rules. Having one simple auction type would apparently be a much better strategy to achieve high-volume end-consumer use than would offering an array of customized types.

[4] Onsale (`www.onsale.com`) does offer a downloadable bid monitoring application ("BidWatch") based on this approach.

is triggered not by the passage of time (mediated by an auctioneer watching the clock), but is rather event-driven by receipt of bid messages.

2. As a sealed-bid auction, where the auctioneer reveals selected information about the bidding status at selected (possibly event-driven) times.[5]

For some price-revision and information-revelation policies, these views correspond to identical auction processes. We therefore find little loss in generality by focusing on sealed-bid mechanisms exclusively, as we can effectively replicate the behavior of outcry auctions in all essential aspects through suitable revelation policies. Note that although we wish to avoid all requirements for strong synchronization, some clock-triggered events (e.g., scheduled notifications or auction closings, or those based on inactivity periods) are essential.

The popular Internet auctions typically attempt to convey the feel of an outcry auction through what is actually an iterative sealed-bid mechanism. The auctions adopt a near-universal policy of displaying the current highest bid (most, but not all, including some identifying information regarding the bidder)—that is, the bid that would win if the auction were to end at that moment—or a reserve price if there are no bids. Thus, at any point the state of the auction resembles a state within the familiar ascending (English) open outcry process.

It is the progression of states that is different, and varies. In all of these auctions, the state changes on receipt of a higher bid by another bidder (perhaps required to exceed the current bid by some minimum increment δ). As in the English auction, the previous high bidder (or anybody else) would then have to bid again to recover its top spot. To compensate for the fact that these bidders may no longer be attending, some Internet auctions (e.g., Onsale)—especially those run by the sellers—send e-mail messages to those outbid.

An alternate approach[6] is to attempt to elicit from participants their "maximum bids", that is the limit of how high they would go if they attended to the auction through its whole course. These auctions then promise not to reveal these maximums, but only the minimum amount necessary to outbid the next highest bidder. Interestingly, the auctions' on-line documentation tends to explain these rules in terms of a simulated real-time outcry auction. For example, Auction Universe refers to an "auto-bidding" feature called "RoboBid" that monitors the process and automatically places incremental bids when necessary, up to the specified maximum amount. This process leads to a result equivalent to a sealed-bid process, where the price paid is that of the second-highest bid, plus the increment δ. Thus, under a well-known auction analysis [2], this approach corresponds exactly to the English auction, which in turn is considered strategically equivalent (modulo δ) to the second-price or Vickrey auction [8]. Since many bidders do not realize (or believe) that bidding one's true valuation is a dominant policy—as follows from the standard analysis—these auctions also tend to notify participants when they are outbid. (One wonders whether their actual implementations employ "RoboBid", since they could more directly obtain the result without iteration.) Note that a seller-run auction may have more difficulty in credibly promising bidders that their maximum-bid information will not be exploited. This would explain why this approach is taken by third-party auction sites, but not seller-run sites, in general.[7]

4 Enabling Asynchrony

Auction processes are defined by their significant events. Bid events are controlled by participants; information revelation and closing the market (called a *clear*) is controlled by the auction. When an event happens can be as important as what happens, and so it is essential that the auction maintain temporal consistency during its operation.

The Internet medium poses some particular difficulties with respect to temporal consistency. Of course, the auction has no control over transmission delays, so bid messages and price notifications can take arbitrary time. Moreover, processing auctions can be complex (depending on the number and size of auctions, and the auction rules), yet the auction must remain available to bidders at all times.

In our design of the AuctionBot, we have therefore decoupled the bidder interface from the auction processing completely. The bidding interface is always available through an HTTP server (or a TCP server for

[5] Of course, one might question whether a mechanism that reveals information can be properly termed "sealed bid". The point is that the auction controls exactly what information is revealed, and this typically is far less than the complete information state. For example, as we see below, auctions often do not reveal actual prices specified by bidders. We adopt this relaxed notion of a sealed-bid auction in the following discussion.

[6] Taken by auctions including Ebay (`www.ebay.com`), Auction Universe (`www.auction-universe.com`), and several others.

[7] Onsale has recently introduced a client application that performs this same function. A client could keep the information away the from the server, but it is unclear whether there is any difference in surety about privacy between information residing on the server or in a client that is distributed by the auction vendor.

software agent interactions). Meanwhile, a scheduler process continually monitors the database for auctions that have events to execute or bids to verify.[8] When it finds such an auction, it forks the auctioneer program corresponding to the auction's type.

The advantage of this decoupling is robustness. If the scheduler goes offline or falls behind in its tasks, the interface continues to operate, and vice versa. All data is timestamped so that when an auction event occurs, it does so with respect to the set of bids that were active at the time the event was supposed to happen.

The asynchrony between the scheduler and the interface, however, creates some interesting bookkeeping challenges. The general problem is to ensure that only the correct bids are in the active state at a given auction event. For example, Figure 1 depicts a situation in which an agent bids before the time of a scheduled clear and then places a revised bid after the scheduled clear. The scheduler is slightly delayed and does not invoke the auction until after the second bid has been placed. This could not happen if the auction were running synchronously. However, decoupling the AuctionBot interface and scheduler necessitates that we take express measures to ensure that the clear executes with bid A and not B active.

The dependency between the validation of bids and scheduled clear events goes both ways. Consider Figure 1 again, but this time suppose it illustrates an auction in which the scheduled clear is based upon inactivity, and that the period of inactivity is greater than the time interval between A and B. The scheduled clear time indicated in the diagram corresonds to the inactivity calculated from a bid previous to A. If the delayed scheduler does not validate bid A before the clear time indicated, it might inappropriately fail to reset the inactivity clock. The correct behavior is to extend this period beyond the time of bid B, thus including the bid.

To address these issues we need to keep careful track of the state of a bid. Figure 2 diagrams the state transitions that a bid might undergo over its lifetime. The auctioneer uses only active bids, indicated by gray boxes, when making clear or price-quote calculations. The interface inserts a new bid into the database with a timestamp and a state of `unprocessed`. The next time it runs, the auctioneer first determines when its next clear or quote event should occur. It then loads all the bids submitted before the time of the next event. If the bid state is `unprocessed`, the auctioneer verifies whether the bid satisfies the auction's rules, and changes the bid state to either `valid` or `rejected`. The auctioneer must also check whether any active bids have expired or been replaced, and move them to the appropriate closed state. We track five distinct types of closed states, for auditing purposes.

Bid withdrawals, like submissions, arrive asynchronously, and can occur out of sequence with clear events. To track a user's request to withdraw a bid, we include two extra states. When a user withdraws a bid, indicated by the dashed arrows, the interface records the appropriate `withdrawrequested` state (qualified by whether it is unprocessed or valid), and marks it with a timestamp. If the withdraw request occurred before the next auction event, the auctioneer moves the bid to the `withdrawn` state. Otherwise the auctioneer considers the bid active at the time of the auction event.[9]

5 Timing and Performance

Even without a strict requirement for synchrony, the timing of auction processing can have a signficant influence on the performance of the system, and the outcome of the auction. If the scheduler falls significantly behind the interface, participants experience a noticeable delay in price quotes, and so are forced to act with outdated information. In an iterative bidding process, the delay will extend the latency of the overall auction in proportion to the number of iterations. If clearing times are based on inactivity periods, and participants bid based on changing prices, then delays in price quotes can cause the auction to close prematurely.

This issue is particularly important in the context of automated trading agents. Software agents can submit bids and information requests at high frequency, thus imposing arbitrarily high loads on an auction server. For this reason, online auctions are quite vulnerable to denial-of-service attacks, as some participants in the auction may have an interest in delaying information to or inhibiting access by the other particpants.

6 Conclusions

Internet auctions present many interesting design choices, including several determining the temporal course of events. We have argued that attempting to mimic the real-time open outcry character of familiar offline auctions should not be a design goal, as it would present difficult technical challenges for the Internet

[8] We perform some simple validation through the interface, but some auctions may have complex eligibility rules that depend on the state of the auction and its existing bids. Validating bids offline eliminates a potential bottleneck at the interface.

[9] This may involve verifying the bid and changing it from `withdraw-requested(unprocessed)` to `withdraw-requested(validated)`.

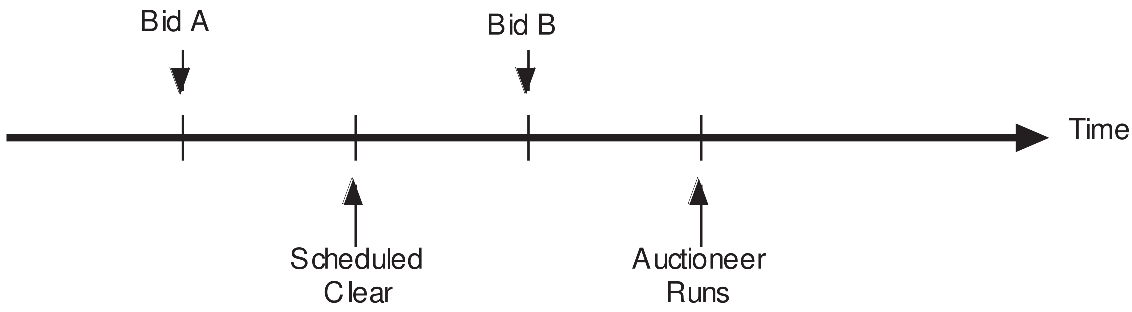

Figure 1: Example of bids before and after a scheduled event.

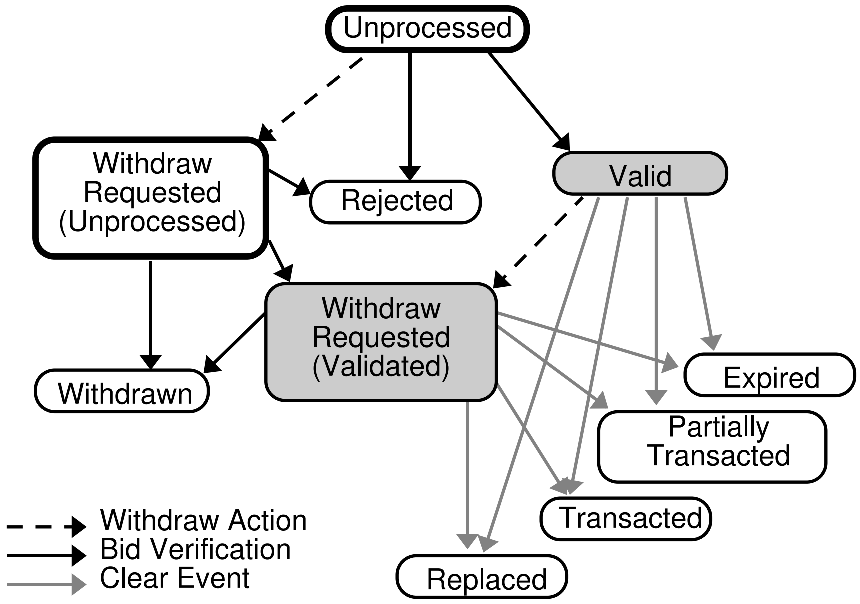

Figure 2: State transition diagram for bids.

medium, and is generally disadvantageous for participants anyway. Instead, we have focused on issues of maintaining temporal consistency through robust management of asynchronous protocols. With reasonable care, we can ensure that auction events execute with their appropriate information states, and thus exercise reliable control of the auction process.

The problem of ensuring timeliness of auction information is significantly more challenging. Any server will have a maximum bid-processing rate, and so potential demand above that rate can introduce delays. Although we can ensure that these delays do not compromise the integrity of the auction rules, they still may have consequences for the latency and outcome of the auction process.

Acknowledgments

This work was supported in part by grants from NSF, AFOSR, and DARPA, and an equipment donation from IBM. We are grateful to the many University of Michigan students who have contributed to the development of the AuctionBot, and to those who have participated in AuctionBot trials.

References

[1] Daniel Friedman and John Rust, editors. *The Double Auction Market.* Addison-Wesley, 1993.

[2] R. Preston McAfee and John McMillan. Auctions and bidding. *Journal of Economic Literature*, 25:699–738, 1987.

[3] R. Preston McAfee and John McMillan. Analyzing the airwaves auction. *Journal of Economic Perspectives*, 10(1):159–175, 1996.

[4] Paul Milgrom. Auctions and bidding: A primer. *Journal of Economic Perspectives*, 3(3):3–22, 1989.

[5] Tracy Mullen and Michael P. Wellman. Market-based negotiation for digital library services. In *Second USENIX Workshop on Electronic Commerce*, pages 259–269, Oakland, CA, 1996.

[6] Juan A. Rodríguez-Aguilar, Francisco J. Martín, Pablo Noriega, Pere Garcia, and Carles Sierra. Competitive scenarios for heterogeneous trading agents. In *Second International Conference on Autonomous Agents*, pages 293–300, Minneapolis, 1998.

[7] Lyle H. Ungar, David C. Parkes, and Dean P. Foster. Cost and trust issues in on-line auctions. In *Agents-98 Workshop on Agent-Mediated Electronic Trading*, pages 161–172, Minneapolis, MN, May 1998.

[8] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[9] Peter R. Wurman, William E. Walsh, and Michael P. Wellman. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, to appear.

[10] Peter R. Wurman, Michael P. Wellman, and William E. Walsh. The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In *Second International Conference on Autonomous Agents*, pages 301–308, Minneapolis, 1998.