AI Magazine Volume 23 Number 3 (2002) (© AAAI)

Specifying Rules for Electronic Auctions

Peter R. Wurman, Michael P. Wellman, and William E. Walsh

■ We examine the design space of auction mechanisms and identify three core activities that structure this space. Formal parameters qualifying the performance of core activities enable precise specification of auction rules. This specification constitutes an auction description language that can be used in the implementation of configurable marketplaces. The specification also provides a framework for organizing previous work and identifying new possibilities in auction design.

Economics is fundamentally about the allocation of scarce resources. Given that many multiagent systems involve the allocation of resources, it is natural that the connection between AI and economics has become a common theme in AI. This emphasis is also certainly influenced by the automation of commercial activities on the internet and the potential benefits of intelligent software support for these economic activities.

Auctions are central to this confluence of research agendas because they represent a class of basic mechanisms by which economic systems compute the outcome of social interactions. At the same time, as a key component of business-to-business internet marketplaces, auctions pose computational and engineering challenges both in the design of the auction servers and in the construction of software to support the decision tasks of auction participants.

In economic literature, analysis of auction protocols sometimes blurs the distinction between the "rules of the game" (hereafter called the *mechanism*) and the behavior of the participants. This orientation is natural because the goal of the analysis is to predict the performance of a particular mechanism in a particular setting. A large body of literature exists mapping combinations of mechanisms and assumptions about participant types to outcomes (Engelbrecht-Wiggans 1980; Friedman 1993; McAfee 1992; Milgrom and Weber 1982).

However, as computer scientists interested in building auction systems, we take an operational perspective on this corpus of knowledge. Assumptions about the types of participants are critical to the analysis of an economic model and the selection of a particular mechanism. However, they are largely irrelevant to the operation of a particular mechanism and often unknown to the designers of generic auction platforms.

This article has several goals: First, we draw a sharp distinction between the mechanism and the participants and discuss the latitude that a designer might have depending on the scope of control. Second, we present an operational parameterization of the space of auction designs that lends itself to modular implementation in servers and precise descriptions of auction rules. Finally, we want to convey a sense of the complexity of the auction environment and the challenges associated with designing agents capable of participating in markets.

System, Auction, or Agent

Negotiation frameworks can broadly be classified by the manner in which the participants communicate. In *unmediated negotiation,* the participants send messages directly to each other. *Mediated mechanisms* involve a third party who manages communication among the participants. The mediator is more than a mes-

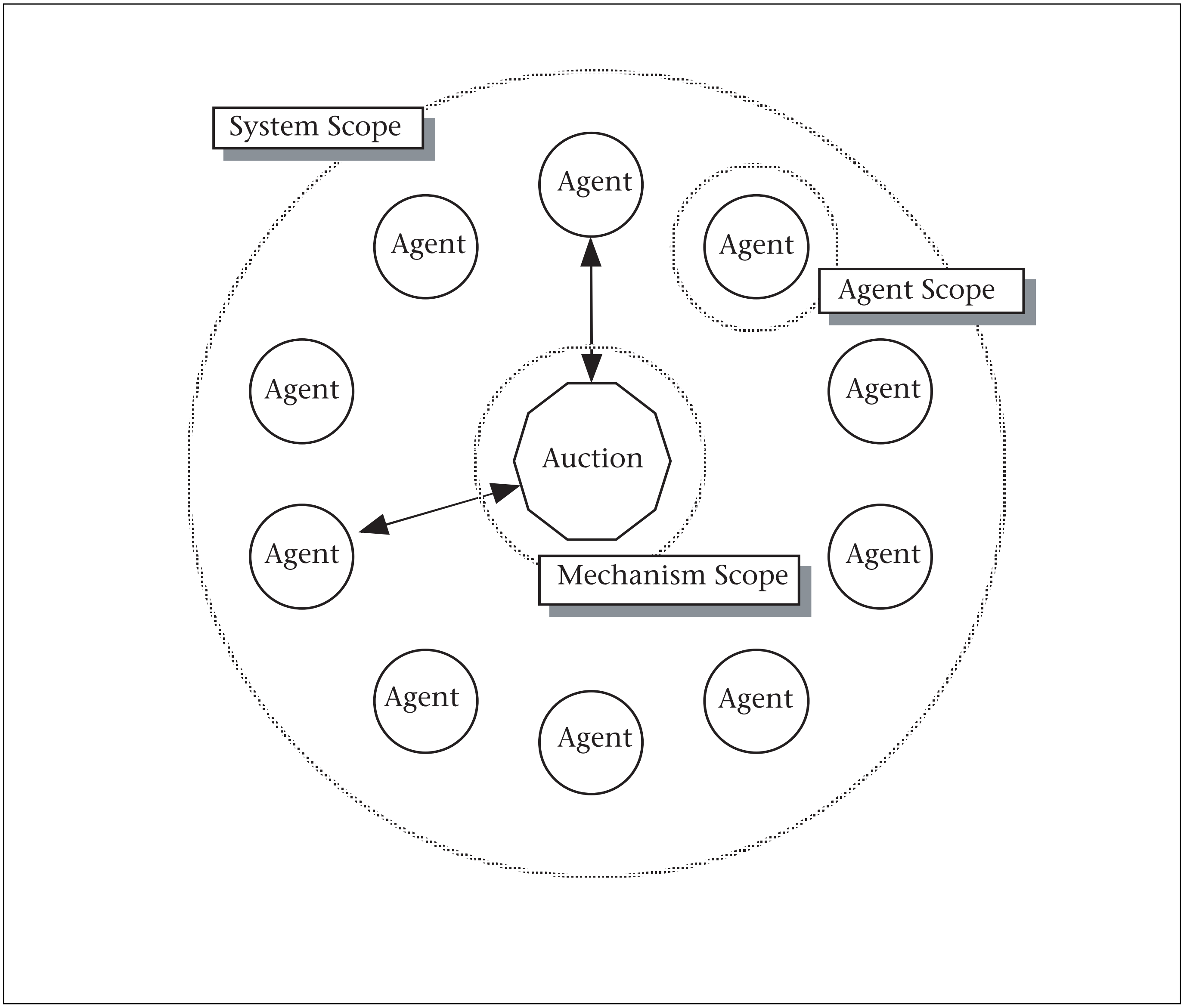


Figure 1. Three Scopes of Concern in Mediated Negotiation.

sage router; by determining what information is reported to whom and when as a function of messages received, it defines the rules of the game and ultimately determines the outcome of the negotiation.

Auctions are mediators that facilitate the negotiation of market-based exchanges. The auction accepts messages in the form of bids, which express a willingness to exchange particular quantities of resources for specified monetary values. This definition includes a wide variety of institutions in the modern economy, including used car auctions, art auctions, the stock market, and the competitive bidding process used by companies and governments to contract suppliers. Auctions have also been used to mediate resources in multiagent systems (Durfee, Kiskis, and Birmingham 1996; Tsvetovatyy 1997).

When analyzing any multiagent system involving negotiation, we must be very careful to clearly state which elements of the system are under the control of the designer. Figure 1 illustrates the following three cases:

First is *agent scope:* The designer controls a single agent. Generally, the designer will be concerned with maximizing the individual agent's utility. The agent designer does not get to choose the rules of the game but will instead look for ways to manipulate the game to the benefit of his/her agent.

Second is *mechanism scope:* The designer controls the mechanism but not the agents that participate in it. The designer's task is one of incentive engineering—to select rules for the game that induce the agents to play in a manner that leads to the desirable social objectives. Typically, the objective is to maximize the overall quality of the allocation or achieve some goal of a particular agent with authority over the mechanism (for example, maximizing revenue to a distinguished seller).

Third is *system scope:* There are some situations in which the designer has control over both the mechanism and the agents. In such cases, the designer has the freedom to choose mechanisms and agent strategies in a way that satisfies the design objectives.

The third scope applies in closed systems in which a single entity is controlling all elements of the system. One example is a company installing an agent-based factory control architecture in which the agents represent jobs and machines and, through distributed computation, negotiate an allocation of machine time to jobs. Because the designer controls all elements of the system, weaknesses in the selected auction mechanism can be compensated for in the agent strategies. However, if the agent behaviors imposed are obviously irrational, then one might question whether the system is best viewed in agent terms at all.

Most electronic-commerce applications entail open systems; each participating entity is in control of its own software representative. The mechanism designer can define the communication interfaces and select the mechanism's rules but cannot enforce particular strategic behaviors. If there is real money at stake, participants will search for strategies that earn them a greater share of the pot.

A designer working within any of these scopes is necessarily concerned with the rules of the auction.

Parameterized Auction Rules

In the process of designing the MICHIGAN INTERNET AUCTIONBOT (Wurman, Wellman, and Walsh 1998), we studied many auctions described in the literature and used on the world wide web. In attempting to characterize their similarities and differences, we found it convenient to organize their features according to how they perform three basic activities common to all auctions.

First is handle bid requests. The messages agents send to the auction typically constitute offers to participate in deals at specified terms. We assume that agents can have at most one active offer, or *bid,* in the system at any particular time. This assumption is without loss of generality because the framework admits arbitrarily complex bids. A bid can be replaced simply by submitting a new bid. A *withdraw* is simply a request to turn an active bid into a null bid.

Second is compute exchanges. The outcome of an auction is a set of deals, or *exchanges,* consistent with the offers expressed in the bids received. The task of computing exchanges is called *clearing* because it typically leaves no possible exchanges among the remaining bids.

Third is generate intermediate information. Many auctions reveal information about the state of bidding during the process, intended to help guide bidders toward a final outcome. Often, the information takes the form of hypothetical prices summarizing the potential deals implicit in the current bid state.

In the rest of this section, we provide an overview of the range of policies governing each of these tasks, followed by a discussion of the scheduling of the tasks. A mathematical treatment of these rules is provided in Wurman, Wellman, and Walsh (2001).

It should be noted that the parameterization is defined over *multicommodity auctions,* that is, auctions whose purview might be more than one type of resource. Thus, the parameterization can usefully be applied to the recent work in combinatorial auctions (auctions that permit bids on combinations of items). In fact, it was the attempt to map common auction rules to these more complex domains that led to many of the insights in this work.

Bidding Rules

The *bidding rules* determine the semantic content of messages, the authority to place certain types of bids, and admissibility criteria for submission and withdrawal of bids.

In an auction for a single item, the content of a bid is simply a *price*. In multiunit auctions, the bid might specify a schedule of prices and quantities indicating the amount of the resource that the bidder is willing to buy or sell at every price. In many cases, complex bid expressions can be simplified by allowing bids to be divisible. For example, the expression "I will sell up to 100 units at $12 apiece" is much more concise than listing all possible quantity values between 1 and 100. More generally, a *bid* is a collection of offers, each of which specifies a monetary value and a combination of resource quantities. This general definition allows for expressions such as "I am willing to pay $10 to supply one unit of A and receive two units of B."

An auction might have rules that specify

Agent	Offer	Submission Time
a_6	Buy 1 unit at $6	t_1
a_5	Sell 1 unit at $5	t_3
a_4	Buy 1 unit at $4	t_4
a_3	Buy 1 unit at $3	t_6
a_2	Sell 1 unit at $2	t_5
a_1	Sell 1 unit at $1	t_2

Table 1. An Example Six Bids.

which bidders are authorized to make which types of bids. A *classic English auction* is an auction in which only the designated seller can place a sell offer (which determines the reserve price). In many B2B procurement auctions, suppliers go through a prequalification process that grants them the authority to participate. In addition, in some B2B auctions, bidders might be assigned to classes, some of which might receive special consideration by the auction. In procurement situations, the buyer is able to favor its most preferred suppliers while it fosters a dynamic and competitive marketplace.

The rules that govern bid admission and withdrawal are more complex. *Single-sided auctions* are usually designed with rules that require bidders to improve their bids as the auction progresses. The English auction achieves its ascending nature by requiring that a new bid be higher than the current highest bid, typically by some increment. In the more general multicommodity case, we find it useful to satisfy this intent by combining two rules.

First is *bid dominance:* This rule requires that an agent's replacement bid be an improvement over its previous bid. Typically, offers to sell (buy) improve by offering to sell (buy) at least as many units at each price.

Second is *beat the quote:* This rule requires that an agent's replacement bid cause it to be more active—buying or selling more—than its previous bid. Unlike the bid dominance rule, the evaluation of activity is with respect to the price information revealed by the auction (price quotes are discussed in more detail later).

The beat-the-quote rule is more directly inspired by the English auction, but it is not sufficient for complex settings. In multiunit auctions for a single resource in which agents submit bid schedules, it is often desirable to require not only that an agent's revised bid increase its activity level but also that the agent not decrease its demand (or increase its supply) at other prices. Likewise, in the multicommodity case, it is usually too restrictive to require that an agent not decrease its activity level on any commodity from one bid to the next. Such a rule would, for example, prevent an agent that is currently winning *A* from placing a bid that would cause it to win *B* and not *A*. For such situations, bid dominance combined with a beat-the-quote rule that requires increased activity on at least one resource (but permits decreased activity on others) might be the right combination.

In many auctions, participants can withdraw bids or submit bids that are valid for only a specified period of time or until the next clearing event. Although most auctions either allow withdrawals or not, some auctions can permit bid withdrawals only if the bid is not currently winning.

Activity rules—a recent innovation in multicommodity auction design—are also motivated by the desire to encourage progressive bidding. Such rules typically restrict a bidder's allowable actions based on its activity level, which might be measured by what it would hypothetically exchange in the current state or directly in terms of its bidding history. The Federal Communications Commission spectrum license auctions (McMillan 1994; Milgrom 2000) are a well-known auction with activity rules. To our knowledge, the range of choices of activity rules has not been explored systematically.

Clearing Rules

As noted earlier, the principal task of an auction is to compute exchanges based on bids received. Generically, this task involves determining (1) which agents trade and (2) what payment is associated with each transaction. Many policies have been proposed for computing the exchange set and the transaction payments. We generically refer to these policies as *matching functions* and identify some key attributes and candidate policies.

A *bidder's surplus* is the difference between the amount that he/she is willing to pay (accept, if selling) and the actual monetary amount of the transaction. If a transaction generates nonnegative surplus for both agents, it is considered mutually beneficial. A matching function is locally efficient if it produces transactions that generate as much surplus—with respect to willingness expressed in the bids—as any other

Exchange	Trades	Total Surplus	Efficient	Uniform Price
e_1	$\{a_1 \rightarrow a_6,\ a_2 \rightarrow a_4\}$	7	Y	Y
e_2	$\{a_1 \rightarrow a_4,\ a_2 \rightarrow a_6\}$	7	Y	Y
e_3	$\{a_1 \rightarrow a_6\}$	5	N	Y
e_4	$\{a_1 \rightarrow a_4,\ a_2 \rightarrow a_3,\ a_5 \rightarrow a_6\}$	5	N	N
e_5	$\{a_1 \rightarrow a_3,\ a_2 \rightarrow a_4,\ a_5 \rightarrow a_6\}$	5	N	N
e_6	$\{a_1 \rightarrow a_4,\ a_2 \rightarrow a_3\}$	4	N	Y
e_7	$\{a_1 \rightarrow a_3,\ a_2 \rightarrow a_4\}$	4	N	Y
e_8	$\{a_1 \rightarrow a_4,\ a_5 \rightarrow a_6\}$	4	N	N
e_9	$\{a_2 \rightarrow a_6\}$	4	N	Y
e_{10}	$\{a_2 \rightarrow a_4,\ a_5 \rightarrow a_6\}$	3	N	N

Table 2. Some of the Feasible Exchange Sets for the Six-Bid Example.

consistent exchanges. Generally, there are many ways to divide the surplus among the agents for any given locally efficient allocation.

Local efficiency serves as a proxy for global efficiency when the auction has only the bids on which to base its computation.

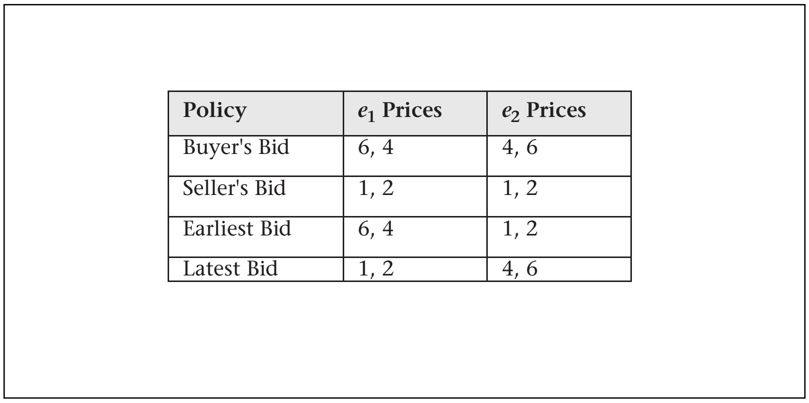
Throughout this section and the next, we present examples of a single-commodity auction with multiple buyers and sellers who place divisible bids. Because of space limitations, we provide only an overview of the application of these concepts to more complex trading scenarios. We use the set of bids shown in table 1 to illustrate some common policies in the single-commodity, divisible-bid case.

The set of mutually beneficial individual trades is $a_1 \rightarrow a_3$, $a_1 \rightarrow a_4$, $a_1 \rightarrow a_6$, $a_2 \rightarrow a_3$, $a_2 \rightarrow a_4$, $a_2 \rightarrow a_6$, and $a_5 \rightarrow a_6$. Of course, some of these trades are mutually exclusive. A subset of the feasible combinations of trades is shown in table 2. The table is augmented with a column indicating what total surplus was created by the set of exchanges, whether the exchange set is locally efficient, and whether it is supportable by uniform prices (described later).

A general procedure for finding the set of bids that entail the locally efficient allocation when resources are discrete and bids are divisible is as follows: Rank all the bids (both buy and sell offers) by bid value. Let M be the number of unit sell offers in the bid set. Counting down from the highest bid in the ranked set, identify the Mth and (M + 1)st bids and their respective prices, p_M and p_{M+1}. Let m be the number of sell offers at or below p_{M+1} and n be the number of buy offers at or above p_M. Let $l = \min(m, n)$. The set of bids that belong in the exchange set is then the l highest buy offers and the l lowest sell offers. This procedure, along with an analysis of its implications in sealed bid auctions and an efficient algorithm to implement it in iterative auctions, is presented in Wurman, Walsh, and Wellman (1998).

Applying this procedure to the example in table 1 produces $M = 3$, $p_M = 4$ and $p_{M+1} = 3$. The set of buyers at or above p_M is $\{a_4, a_6\}$, and the set of sellers at or below p_{M+1} is $\{a_1, a_2\}$. In table 2, the exchanges e_1 and e_2 represent the two permutations of the members of the locally efficient exchange set.

In the example, all bids are for a single unit. The procedure also works when multiunit bids are divisible. When bids are indivisible, we are generally faced with a packing problem. In practice, auction sites such as UBid use a greedy algorithm to compute the exchange set. Recently, computer scientists have investigated algorithms to solve the *winner determination problem* (that is, compute the locally efficient



Policy	e_1 Prices	e_2 Prices
Buyer's Bid	6, 4	4, 6
Seller's Bid	1, 2	1, 2
Earliest Bid	6, 4	1, 2
Latest Bid	1, 2	4, 6

Table 3. Prices Paid under Four Discriminatory Policies.

combination of bids) in combinatorial auctions (Andersson, Tenhunen, and Ygge 2000; Fujishima, Leyton-Brown, and Shoham 1999; Sandholm et al. 2001).

Matching policies can also be classified by the manner in which they set the transaction price(s). Clearly, the price associated with any particular transaction must be no more than the buyer's bid and no less than the seller's bid. A uniform price policy uses the same price for all trades computed during a single clear event. In contrast, discriminatory policies can charge different prices to different agents. This discrimination is not necessarily tied to the agent's identity; rather, it is often induced by attributes of the agent's bid.

In exchanges e_1 and e_2, any price p such that $4 \geq p \geq 2$ satisfies the conditions of being a uniform price for both transactions. Most uniform price policies, however, select an *equilibrium price*. In the single-unit case, p_M and p_{M+1} define the range of equilibrium prices. By picking a price within this range, we guarantee not only that the agents in the exchange set are willing to trade at the computed price but also that the agents not in the exchange set do not want to trade.

Other pricing schemes are present in the literature that generate uniform prices but do not use the locally efficient exchange set. For example, McAfee's (1992) DUAL PRICE mechanism computes an equilibrium price but then removes the lowest buyer and highest seller from the exchange set if one of their bids helps determine the transaction price. The dual-price policy would result in exchange set e_3. This scheme sacrifices efficiency to ensure that truth telling is a weakly dominant strategy (that is, an agent could do no better than to place an offer representing its true value). Although McAfee studied only a sealed- bid version of the auction, the matching function can be isolated as a component in the parameterization framework.

Under discriminatory pricing, the price of each transaction is bounded only by the bids involved. Several methods present themselves in this case. For example, we can set the price of each transaction according to the bid of the buyer, or always by the bid of the seller, or by some linear combination thereof. An alternative is to base the price on the order the bids were received typically but not necessarily, using the price of the earlier bid. Table 3 shows the prices resulting from each of these policies when applied to exchange sets e_1 and e_2. The prices are presented in the order of the two transactions in the respective set.

Notice that the payment that an individual agent makes (receives) depends on which of the two exchange sets is used. Under the *buyer's-bid policy*, a_1 receives 6 if the auction uses exchange set e_1 but only 4 in e_2. This example illustrates a key difference between the uniform price and discriminatory policies; in both cases, an agent might be able to manipulate its bid to affect the price it pays, but discriminatory pricing can also give incentives to the agents to consider the impact of their bid on their pairing.

When indivisible bids for multiple units are permitted, equilibrium prices might not exist. Most, if not all, internet auction sites that permit indivisible bidding sidestep this problem by adopting a buyer's-bid pricing policy.

The multicommodity case has even more degrees of freedom and trade-offs that can be made among desirable properties. For a more detailed discussion of combinatorial auctions and the manner in which they fit into the parameterization framework, see Wurman, Wellman, and Walsh (2001).

Tie-breaking rules are the final component of the matching function that must be specified. Common rules for breaking ties include random choice, in favor of the earliest bid, or in favor of the bid with the highest quantity.

Before leaving the subject of transactions, we should note that there are several methods by which an auctioneer can generate income for itself. An auction can charge one or more of the following: an entrance fee, a fee for a bid, or a percentage of the transaction price either at a fixed rate or at a rate that is a function of the transaction quantity. These fees can be levied against the buyer or the seller.

Information Rules

Many auctions reveal intermediate information to the bidders (auctions that do not reveal any information are designated sealed bid). This information can come in several forms:

Order book: An *order book* is the paper-based

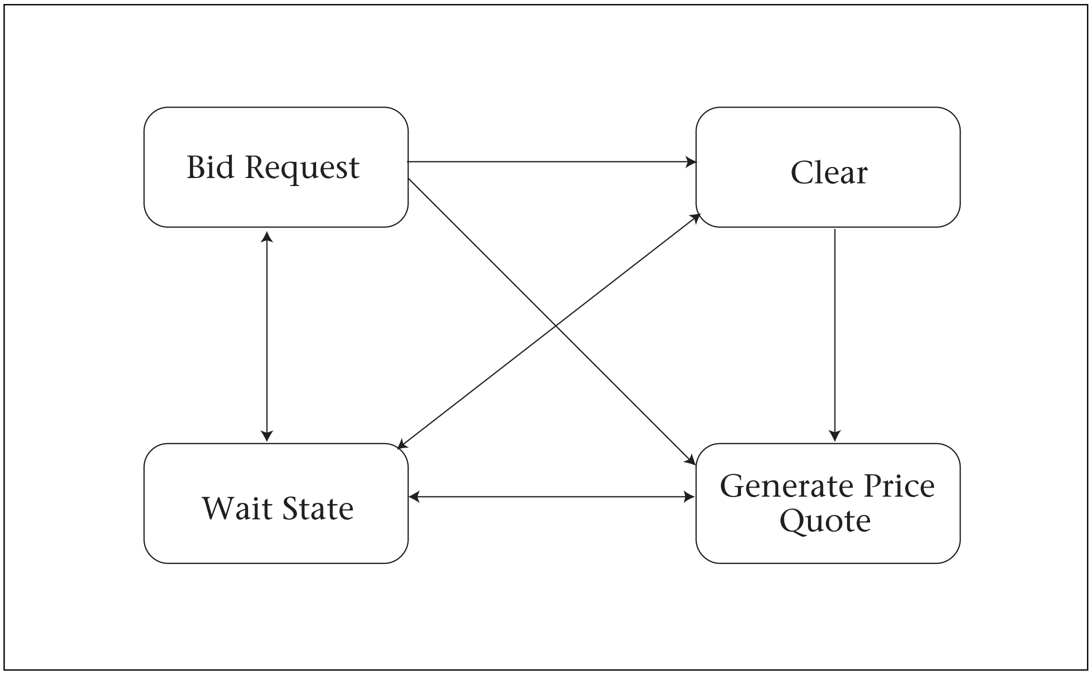


Figure 2. The Possible Sequences among Auction Activities.

method traditionally used on the floor of the New York Stock Exchange to keep track of bids. We use the term more generally to refer to a list of the current bids. Many online auction sites have a policy of revealing all the bids. Alternatively, the auction can reveal the bid values, the quantities, or the bidders' identities.

Transaction history: An auction can publicize information about past transactions, including the prices, quantities, or identities of the traders.

Price quotes: A price quote serves two purposes. First, it informs bidders of the range of bids that would have been in the exchange set at the time of the quote. Second, it guides bidders to make new bids that will become part of the exchange set.

The rest of this section is dedicated to discussing some of the features of price quotes. We refer to the policy that an auction uses to compute price information as the *quote function.* Often the quote function is simply an impotent version of the matching function. To generate a quote, the auction simply runs the matching function and announces the computed prices but does not announce the transactions.

For example, when bids are divisible and when resources are discrete, the procedure described earlier to compute equilibrium prices for the transactions can also be used to compute quotes. The prices p_M and p_{M+1} are referred to as the *ask quote* and *bid quote*, respectively.

Notice that in the example, the information provided by the bid-ask quote satisfies both of the properties of price quotes. First, all buyers who have bid greater than or equal to the ask quote (a_6 and a_4) are currently winning, and all sellers who have bid less than or equal to the bid quote (a_1 and a_2) are currently winning. Note that this is not necessarily so; the result is ambiguous when the bid quote, ask quote, and the agent's bid are all equal.

The quote also correctly informs a new bidder of the range they must bid to become a member of the exchange set. In this case, a new or nonwinning buyer would have to offer at least 4 to displace buyer a_4, and a nonwinning seller would have to offer less than 3 to match with buyer a_3.

We say a quote is *separating* if all agents can correctly determine their tentative allocation. The quote is noisy otherwise. Naturally, the possibility of noisy quotes complicates the application of the beat-the-quote rule. To avoid the potential ambiguity, we require that to satisfy the beat-the-quote rule, a bid must increase the bidder's activity level as entailed by the price quote (in contrast to the bidder's actual activity level). Intuitively, this definition makes sense because we can expect a bidder to improve his/her activity level with respect only to the information provided.

An auction has *anonymous quotes* if it reports the same quote to each agent. The alternative is to customize quotes for each agent. When anonymous quotes are ambiguous, we can complement the message with the agent's exact tentative allocation without resorting to discriminatory pricing.

When multiunit, indivisible bids are allowed, separating prices might not exist. In such cases, we need to resort to nonlinear pricing (let the price for each unit vary with quantity).

The issues become more complex in the multicommodity case. In Wurman and Wellman (2000), we showed that a lattice of separating bundle prices always exists in one-sided combinatorial auctions. Our A1BA mechanism is based on this method of constructing prices. IBUNDLE is a combinatorial auction that permits discriminatory bundle pricing (Parkes 1999). RSB (Rassenti, Smith, and Bulfin 1982) and RAD (DeMartini et al. 1998) use approximation techniques to generate (noisy) quotes. AUSM (Banks, Ledyard, and Porter 1989) avoids generating price quotes by revealing the order book.

Sequencing Auction Tasks

The enforcement of bidding rules occurs when the auction receives a bid. However, the generation of price information and the computation of exchanges can be triggered in a variety of ways. The result is that an auction can interleave these three activities in complex ways governed by the associated rules and the sequence of bids received. Figure 2 illustrates the possible transitions between activities. A specific auction will be represented by a subset of this graph, depending on the particular rules.

Clear events can be initiated by the admittance of a new bid, tied to a fixed schedule, or triggered randomly or by a lack of bidding activity. Similarly, the generation of price quotes can immediately follow a clear event or the admission of a new bid or can occur on a fixed schedule, at random times, or after a period of inactivity.

In addition, some auctions are organized into rounds. The parameters described herein precisely specify the rules of each round, and activity rules link the rounds together by tying permissible actions in one round to outcomes or actions in previous rounds.

Finally, the auction design must specify the logical conditions that close the auction. Auctions can close at a fixed time, after a period of inactivity, at the time of a transaction, or at a random time.

Although the choices controlling timing seem relatively minor compared to issues such as which matching function to use, they can have a tremendous impact on the auction. For example, the versions of the English auction used by eBay and Amazon are essentially the same except that eBay closes at fixed times, and Amazon closes when a period of inactivity expires. The difference in this single rule greatly impacts the bidding strategies used by the bidders (Roth and Ockenfels 2000). In eBay auctions, *sniping*—bidding high at the last moment—is such a common strategy that software programs have been built to facilitate it. In contrast, sniping is of little value in Amazon's auctions.

Concluding Remarks

To a great extent, the rules described herein are orthogonal and can be combined in many permutations. The parameters shed light on the commonality among auction types and provide descriptive structure to the space of designs. This structure provides the basis for a language to communicate auction rules to software agents. We have created such a language in XML and have made it available on a web page.[1]

By charting the design space, we also illuminate some of the darker corners and expose new auction types for inspection. For example, defining price quotes in terms of information content enables us to apply the same policy that is used in the CDA (continuous double auction) to a much broader range of uniform price auctions. Similarly, by reflecting on the purpose of the beat-the-quote rule in the English auction, we are able to generate multicommodity versions of the rules.

The structure also focuses attention on the key elements of the auctions: the matching and quote functions. By separating out the ancillary parameters, we can focus on defining new matching functions for the most general cases (two sided, unrestricted bids, and so on). Special cases are implemented by appropriately setting the supporting parameters. This modularity pays dividends in the auction implementation and is easier to extend over time.

Acknowledgments

This work was supported in part by grants from the National Science Foundation (National Young Investigator Award IRI-9457624), the Air Force Office of Scientific Research (F49620-97-0175), and the Defense Advanced Research Projects Agency (Information Survivability Program grant F30602-97-1-0228) as well as an equipment donation from IBM. We would like to thank Jeff MacKie-Mason, Tracy Mullen, and Daniel Reeves for their valuable comments. We are also grateful to Kevin O'Malley and the many students who have worked on the AUCTIONBOT project over the past few years, and we acknowledge fruitful discussion with industrial partners implementing commercial auction systems. Finally, we thank David Taylor for creating the XML schema implementation of the parameterization.

Note

[1] www.csc.ncsu.edu/faculty/wurman/Auction-xsd/ParamAuction.html.

References

Andersson, A.; Tenhunen, M.; and Ygge, F. 2000. Integer Programming for Combinatorial Auction Winner Determination. Paper presented at the Fourth International Conference on Multiagent Systems, 10–12 July, Boston, Massachusetts.

Banks, J. S.; Ledyard, J. O.; and Porter, D. P. 1989. Allocating Uncertain and Unresponsive Resources: An Experimental Approach. *RAND Journal of Economics* 20(1): 1–22.

DeMartini, C.; Kwasnica, A. M.; Ledyard, J. O.; and Porter, D. 1998. A New and Improved Design for Multi-Object Iterative Auctions. Technical report, 1054, Division of Humanities and Social Science, California Institute of Technology.

Durfee, E. H.; Kiskis, D. L.; and Birmingham, W. P. 1996. The Agent Architecture of the University of Michigan Digital Library. In *Readings in Agents,* eds. M. N. Huhns and M. Singh, 98–108. San Francisco, Calif.: Morgan Kaufmann.

Engelbrecht-Wiggans, R. 1980. Auctions and Bidding Models: A Survey. *Management Science* 26:119–142.

Friedman, D. 1993. The Double Auction Market Institution: A Survey. In *The Double Action Market: Institutions, Theories, and Evidence,* eds. D. Friedman and J. Rust, 3–26. Reading, Mass.: Addison-Wesley.

Fujishima, Y.; Leyton-Brown, K.; and Shoham, Y. 1999. Taming the Computational Complexity of Combinatorial Auctions: Optimal and Approximate Approaches. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, 548–553. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Hoos, H. H., and Boutilier, C. 2000. Solving Combinatorial Auctions Using Stochastic Local Search. In Proceedings of the Seventeenth National Conference on Artificial Intelligence, 22–29. Menlo Park, Calif.: American Association for Artificial Intelligence.

McAfee, R. P. 1992. A Dominant Strategy Double Auction. *Journal of Economic Theory* 56(2): 434–450.

McAfee, R. P., and McMillan, J. 1987. Auctions and Bidding. *Journal of Economic Literature* 25(2): 699–738.

McMillan, J. 1994. Selling Spectrum Rights. *Journal of Economic Perspectives* 8(3): 145–162.

Milgrom, P. 2000. Putting Auction Theory to Work: The Simultaneous Ascending Auction. *Journal of Political Economy* 108(2): 245–272.

Milgrom, P. R., and Weber, R. J. 1982. A Theory of Auctions and Competitive Bidding. *Econometrica* 50(5): 1089–1122.

Parkes, D. C. 1999. IBUNDLE: An Efficient Ascending Price Bundle Auction. In Proceedings of the First ACM Conference on Electronic Commerce, 148–157. New York: Association of Computing Machinery.

Rassenti, S. J.; Smith, V. L.; and Bulfin, R. L. 1982. A Combinational Auction Mechanism for Airport Time Slot Allocation. *Bell Journal of Economics* 13(2): 402–417.

Roth, A. E., and Ockenfels, A. 2000. Last-Minute Bidding and the Rules for Ending Second-Price Auctions: Theory and Evidence from a Natural Experiment on the Internet. NBER Working paper, 7729, Harvard University.

Sandholm, T.; Suri, S.; Gilpin, A.; and Levine, D. CABOB: A Fast Optimal Algorithm for Combinatorial Auctions. In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, 1102–1108. Menlo Park, Calif.: American Association for Artificial Intelligence.

Tsvetovatyy, M.; Gini, M.; Mobsher, B.; and Wieckowski, Z. 1997. MAGMA: An Agent-Based Virtual Market for Electronic Commerce. *International Journal of Applied Artificial Intelligence* 11(6): 501–523.

Wurman, P. R., and Wellman, M. P. 2000. AKBA: A Progressive, Anonymous-Price Combinatorial Auction. In Proceedings of the Second ACM Conference on Electronic Commerce, 21–29. New York: Association of Computing Machinery.

Wurman, P. R.; Walsh, W. E.; and Wellman, M. P. 1998. Flexible Double Auctions for Electronic Commerce: Theory and Implementation. *Decision Support Systems* 24(1): 17–27.

Wurman, P. R.; Wellman, M. P.; and Walsh, W. E. 1998. The MICHIGAN INTERNET AUCTIONBOT: A Configurable Auction Server for Human and Software Agents. Paper presented at the Second International Conference on Autonomous Agents, 9–13 May, Minneapolis, Minnesota.

Wurman, P. R.; Wellman, M. P.; and Walsh, W. E. 2001. A Parameterization of the Auction Design Space. *Games and Economic Behavior* 35(1–2): 304–338.

Peter R. Wurman is an assistant professor of computer science at North Carolina State University. He is codirector of the Electronic-Commerce Program at North Carolina State University and does research on electronic auctions, trading agents, and other topics where AI collides with electronic commerce. His e-mail address is wurman@csc.ncsu.edu.

Michael P. Wellman received a Ph.D. from the Massachusetts Institute of Technology in 1988 for his work in qualitative probabilistic reasoning and decision-theoretic planning. From 1988 to 1992, Wellman conducted research in these areas at the United States Air Force Wright Laboratory. For the past nine years, his research has focused on computational market mechanisms for distributed decision making and electronic commerce. As chief market technologist for TradingDynamics, Inc. (now part of Ariba), he designed configurable auction technology for dynamic business-to-business commerce. He has served as executive editor of the *Journal of Artificial Intelligence Research* and chair of the ACM Conference on Electronic Commerce and has been elected councilor and fellow of the American Association for Artificial Intelligence. His e-mail address is wellman@umich.edu.

William E. Walsh is a research staff member in the Agents and Emergent Phenomenon Group at the IBM T. J. Watson Research Center. He earned his Ph.D. in computer science and engineering from the University of Michigan in 2001 for his work on market protocols for decentralized supply-chain formation and distributed constraint satisfaction. With research interests in automated negotiation for electronic commerce and distributed problem solving, Walsh has also worked on market-based scheduling and infrastructure development for the First Trading Agent Competistion. His e-mail address is wwalsh1@us.ibm.com.

Seventh International Conference On Intelligent User Interfaces

CALL FOR PAPERS

iui 2003

January 12-15, 2003
Miami Beach, Florida
www.iuiconf.org

ORGANIZING COMMITTEE

CONFERENCE CHAIR
David Leake, Indiana University (US)

PROGRAM CO-CHAIRS
Lewis Johnson, USC/Information Sciences Institute (US)
Elisabeth Andre, University of Augsburg (GE)

POSTER AND DEMONSTRATION CHAIR
John Domingue, The Open University (UK)

PAPER ABSTRACTS DUE:
September 30, 2002

FINAL PAPERS AND DEMONSTRATION ABSTRACTS DUE:
October 4, 2002

Sponsors:
IUI is sponsored by ACM through SIGCHI and SIGART in cooperation with AAAI and the British HCI Group.

Corporate Sponsors:
IUI's corporate sponsors include RedWhale Software, Mitsubishi Electric Research Laboratories, and Microsoft Research.

The International Conference on Intelligent User Interfaces is the principal international forum for reporting outstanding research and development in intelligent user interfaces. IUI '03 solicits submissions of original papers, posters, and system demonstrations on significant progress and new directions in all aspects of intelligent user interfaces. In addition to showcasing methods for intelligently reasoning and adapting to fit user needs, IUI '03 aims to highlight interfaces whose focus goes beyond rational "intelligence" to address psychological concerns such as emotion, personality, and motivation.

Topics of interest include, but are not limited to:

- Adaptive hypermedia
- Affective interfaces
- Conversational interfaces
- Evaluations of implemented systems
- Intelligent information and knowledge management systems
- Intelligent support for pervasive computing
- Intelligent systems for multimedia presentation
- Intelligent visualization tools
- Interface agents
- Interfaces for the semantic web
- Knowledge-based approaches to user interface design
- Multi-modal interfaces
- Novel application-specific interfaces and interactions
- Novel interaction paradigms
- Personalization and adaptation of behavior
- Proactive and agent-based paradigms for user interaction
- Support for collaboration in multi-user environments
- User modeling for intelligent interfaces
- User studies of relevant techniques

IUI '03 solicits three types of submissions: full papers, short papers, and demonstration abstracts. Full papers must be at most 8 pages long, short papers at most 3 pages long, and demonstrations 1 page long in the CHI paper format. All submissions will be reviewed by an international program committee. Full details are available from http://www.iuiconf.org.

CALL FOR PAPERS

The Sixteenth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems (IEA/AIE-2003) – Loughborough, UK – June 23-26, 2003

Sponsored by: International Society of Applied Intelligence – Organized in Cooperation with: AAAI, ACM/SIGART, CSCSI/SCEIO, ECCAI, IEE, ENNS, INNS, JSAI, LU, and SWT

IEA/AIE-2003 continues the tradition of emphasizing applications of artificial intelligence and expert/knowledge-based system to engineering and industrial problems as well as application of intelligent systems technology to solve real-life problems. Numerous related topics are considered and are listed on the conference URL http://gradients.lboro.ac.uk/iea2003aie

Authors are invited to submit by email to iea2003aie@lboro.ac.uk (1) a key word listing, and (2) their paper, written in English, of up to 10 single spaced pages, presenting the results of original research or innovative practical applications relevant to the conference. Practical experiences with state-of the art AI methodologies are also acceptable when they reflect lessons of unique value. Shorter works, up to 6 pages, may be submitted as "short papers" representing work in progress or suggesting research directions. Submissions are due by **November 12, 2002**, to iea2003aie@lboro.ac.uk. Additional details may be obtained from the web site http://gradients.lboro.ac.uk/iea2003aie or Dr. Paul Chung, IEA/AIE Program Chair, Department of Computer Science, Loughborough University, Loughborough, Leicestershire LE11 3TU, England, UK; FAX +44 0 1509 211586

General conference information can be sought from the General Chair at the following address: Dr. Moonis Ali, General Chair of IEA/AIE-2003, Southwest Texas State University, Department of Computer Science, 601 University Drive, San Marcos TX 78666-4616 USA; E-mail: ma04@swt.edu