

# Specifying Rules for Electronic Auctions

Peter R. Wurman

Michael P. Wellman William E. Walsh

North Carolina State University  
Computer Science  
Raleigh, NC 27695-7535 USA  
wurman@csc.ncsu.edu

University of Michigan  
Computer Science and Engineering  
Ann Arbor, MI 48109-2110 USA  
{wellman, wew} @umich.edu

July 11, 2002

## Abstract

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

## 1 Introduction

As this themed issue attests, the connection between AI and economics has received a lot of attention recently. This seems a natural occurrence given that many multiagent systems involve the allocation of resources, and economics is fundamentally about resource allocation. It 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 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 [5, 6, 11, 14].

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

## 2 System, Auction, or Agent?

Negotiation frameworks can be broadly 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 message router—by determining what information is reported to 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 a specified monetary value. 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 multi-agent systems [4, 19].

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:

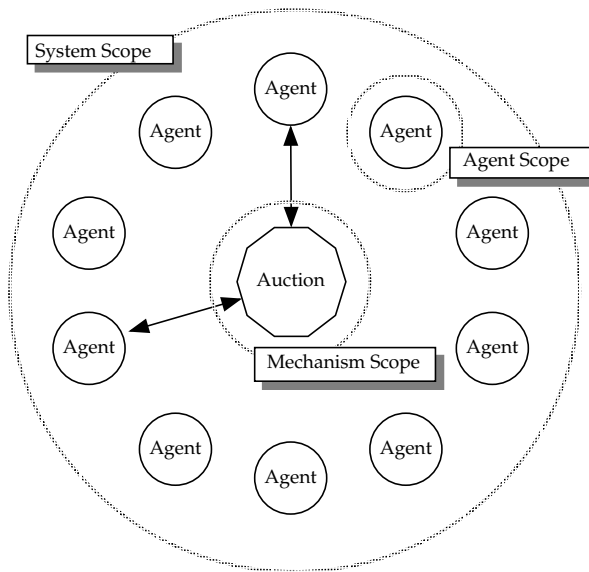


Figure 1: Three scopes of concern in mediated negotiation.

1. **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 her agent.
2. **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 to achieve some goal of a particular agent with authority over the mechanism (e.g., maximizing revenue to a distinguished seller).
3. **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 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 e-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 the above scopes is necessarily concerned with the rules of the auction.

### 3 Parametrized Auction Rules

In the process of designing the Michigan Internet AuctionBot [22], 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.

- **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 request is simply a request to turn an active bid into a null bid.
- **Compute exchanges.** The outcome of an auction is a set of deals, or exchanges, consistent with the offers expressed in bids received. The task of computing exchanges is called *clearing* because it typically leaves no possible exchanges among the remaining bids.
- **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, this 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 the full article [23].

It should be noted that the parametrization is defined over *multi-commodity* auctions, that is, auctions whose purview may be more than one type of resource. Thus, the parametrization can be usefully applied to the recent work in *combinatorial auctions* (auctions which 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.

### 3.1 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 multi-unit 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 instance, 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 allows for expressions like “I am willing to pay \$10 to supply one unit of A and receive two units of B.”

An auction may have rules that specify which bidders are authorized to make which types of bids. A classic English auction can be seen as 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 which grants them the authority to participate. In addition, in some B2B auctions bidders may be assigned to classes, some of which may receive special consideration by the auction. In procurement situations, this enables the buyer to favor its most preferred suppliers while still fostering 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 multi-commodity case we find it useful to satisfy this intent by combining two rules.

- **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.

- **Beat-the-Quote:** This rule requires that an agent’s replacement bid cause it to be more active—buying or selling more—than does 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 below).

The Beat-the-Quote rule is more directly inspired by the English auction, but it is not sufficient for complex settings. In multi-unit 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 multi-commodity 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 instance, prevent an agent that is currently winning A from placing a bid which 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) may 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 may permit bid withdrawals only if the bid is not currently winning.

*Activity rules*—a recent innovation in multi-commodity auction design—are also motivated by the desire to encourage progressive bidding. Such rules typically restrict a bidder’s allowable actions based upon 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 FCC spectrum license auctions [12, 13] are a well-known auction with activity rules. To our knowledge, the range of choices of activity rules has not been systematically explored.

### 3.2 Clearing Rules

As noted above, the principal task of an auction is to compute exchanges based upon bids received. Generically, this 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 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 mutu-

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.

ally 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 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. Due to 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 bids 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 are mutually exclusive. A subset of the feasible combinations of trades are shown in Table 2. The table is augmented with a column indicating the total surplus created by the set of exchanges, whether the exchange set is locally efficient, and whether it is supportable by uniform prices (described below).

A general procedure for finding the set of bids which entail the locally efficient allocation when resources are discrete and bids are divisible is as follows. Rank all of 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  $M$ th 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 earlier work [20].

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

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.

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 multi-unit 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* (i.e. compute the locally efficient combination of bids) in combinatorial auctions [1, 8, 9, 18].

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 may 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 that 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 Dual Price mechanism [10] computes an equilibrium price but then removes the lowest buyer and highest seller from the exchange set. The dual-price policy would re-

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.

sult in exchange set  $e_3$ . This scheme sacrifices efficiency to ensure that truth-telling is a weakly dominant strategy (i.e., 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 parametrization 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 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 upon 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 illustrates a key difference between the uniform price and discriminatory policies; in both cases an agent may be able to manipulate its bid to affect the price it pays, but discriminatory pricing may 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 may 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 multi-commodity case has even more degrees of freedom and tradeoffs that can be made among desirable properties. For a more detailed discussion of combinatorial auctions and how they fit into the parametrization framework, see the full article [23].

Tie-breaking rules are the final component of the matching function that must be specified. Common rules for breaking ties include randomly, 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 a vari-

ety of methods by which an auctioneer generate income for itself. An auction may charge one or more of the following: an entrance fee, a fee per 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 may be levied against the buyer or the seller.

### 3.3 Information Rules

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

- **Order book.** An order book is the paper-based method traditionally used on the floor of the NYSE 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 of the bids. Alternatively, the auction can reveal the bid values, the quantities and/or the bidders' identities.
- **Transaction history.** An auction may publicize information about past transactions, including the prices, quantities, or identities of the traders.
- **Price quotes.** A price quote serves two purposes. First, it inform 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 making 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 for discrete resources, the procedure described above 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 provide 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 in order to become a member of the exchange set. In this case, a new or non-winning buyer would have to offer at least 4 to displace buyer  $a_4$ , and a non-winning 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 solve this, 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 makes sense because we can expect a bidder to improve 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 multi-unit, indivisible bids are allowed, separating prices may not exist. In such cases, we need to resort to nonlinear pricing (let the price per unit vary with quantity).

The issues become more complex in the multi-commodity case. Elsewhere [21], we have shown that a lattice of separating bundle-prices always exist 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 [15]. RSB [16] and RAD [3] use approximation techniques to generate (noisy) quotes. AUSM [2] avoids generating price quotes by revealing the order book.

### 3.4 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 may be triggered in a variety of ways. The result is that an auction may 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 upon 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 or the admission of a new bid, or can occur on a fixed schedule, at random times, or after a period of inactivity.

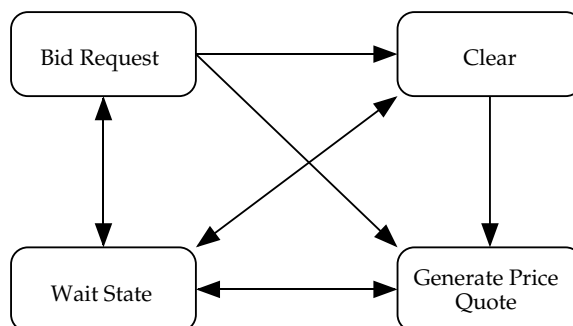


Figure 2: The possible sequences among auction activities.

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, when a transaction occurs, 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 tremendous impact on the auction. For example, the versions of the English auction used by ebay and Amazon are essentially the same except that the former close at fixed times and the later close when a period of inactivity expires. The difference in this single rule greatly impacts the bidding strategies used by the bidders [17]. 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.

## 4 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 from the first author’s web page.

By charting the design space we also illuminate some of the darker corners and expose new auction types for inspection. For example, by defining price quotes in

terms of information content enable us to apply the same policy that is used in the CDA 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 were able to generate multi-commodity 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, etc.). 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 NSF (National Young Investigator award IRI-9457624), AFOSR (F49620-97-0175), and DARPA (Information Survivability program grant F30602-97-1-0228), and 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 parametrization.

## References

- [1] Arne Andersson, Mattias Tenhunen, and Fredrik Ygge. Integer programming for combinatorial auction winner determination. In *Fourth International Conference on Multiagent Systems*, pages 39–46, 2000.
- [2] Jeffrey S. Banks, John O. Ledyard, and David P. Porter. Allocating uncertain and unresponsive resources: An experimental approach. *RAND Journal of Economics*, 20:1–22, 1989.
- [3] Christine DeMartini, Anthony M. Kwasnica, John O. Ledyard, and David Porter. A new and improved design for multi-object iterative auctions. Technical report, California Institute of Technology, November 1998.
- [4] E. H. Durfee, D. L. Kiskis, and W. P. Birmingham. The agent architecture of the University of Michigan Digital Library. In *IEEE Proceedings – Software Engineering*, volume 144, pages 61–71, 1996.

- [5] Richard Engelbrecht-Wiggans. Auctions and bidding models: A survey. *Management Science*, 26:119–142, 1980.
- [6] Daniel Friedman. The double auction market institution: A survey. In Friedman and Rust [7], pages 3–26.
- [7] Daniel Friedman and John Rust, editors. *The Double Auction Market: Institutions, Theories, and Evidence*. Addison-Wesley Publishing, Reading, MA, 1993.
- [8] Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Sixteenth International Joint Conference on Artificial Intelligence*, pages 548–553, 1999.
- [9] Holger H. Hoos and Craig Boutilier. Solving combinatorial auctions using stochastic local search. In *Seventeenth National Conference on Artificial Intelligence*, pages 22–29, Austin, TX, 2000.
- [10] R. Preston McAfee. A dominant strategy double auction. *Journal of Economic Theory*, 56:434–450, 1992.
- [11] R. Preston McAfee and John McMillan. Auctions and bidding. *Journal of Economic Literature*, 25:699–738, 1987.
- [12] John McMillan. Selling spectrum rights. *Journal of Economic Perspectives*, 8(3):145–162, 1994.
- [13] Paul Milgrom. Putting auction theory to work: The simultaneous ascending auction. *Journal of Political Economy*, 108(2):245–272, 2000.
- [14] Paul R. Milgrom and Robert J. Weber. A theory of auctions and competitive bidding. *Econometrica*, 50:1089–1122, 1982.
- [15] David C. Parkes. *iBundle*: An efficient ascending price bundle auction. *First ACM Conference on Electronic Commerce*, pages 148–157, November 1999.
- [16] S. J. Rassenti, V. L. Smith, and R. L. Bulfin. A combinatorial auction mechanism for airport time slot allocation. *Bell Journal of Economics*, 13:402–417, 1982.
- [17] Alvin E. Roth and Axel Ockenfels. Last minute bidding and the rules for ending second-price auctions: Theory and evidence from a natural experiment on the Internet. Technical report, Harvard University, June 2000.

- [18] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. Cabob: A fast optimal algorithm for combinatorial auctions. In *International Joint Conference on Artificial Intelligence*, Seattle, WA, 2001.
- [19] Maksim Tsvetovatyy, Maria Gini, Bamshad Mobsher, and Zbigniew Wieckowski. MAGMA: An agent based virtual market for electronic commerce. *International Journal of Applied Artificial Intelligence*, September 1997.
- [20] Peter R. Wurman, William E. Walsh, and Michael P. Wellman. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, 24:17–27, 1998.
- [21] Peter R. Wurman and Michael P. Wellman. AkBA: A progressive, anonymous-price combinatorial auction. In *Second ACM Conference on Electronic Commerce*, pages 21–29, 2000.
- [22] 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.
- [23] Peter R. Wurman, Michael P. Wellman, and William E. Walsh. A parametrization of the auction design space. *Games and Economic Behavior*, 35(1-2):304–338, Apr.-May 2001.