eMediator: a Next Generation Electronic Commerce Server

Tuomas Sandholm*
sandholm@cs.wustl.edu
Department of Computer Science
Washington University
St. Louis, MO 63130

Abstract

eMediator, a next generation electronic commerce server, demonstrates ways in which AI, algorithmic support, game theoretic incentive engineering, and GUI design can jointly improve the efficiency of ecommerce.

The first component, *eAuctionHouse*, is a configurable auction house that supports a large variety of parameterizable auction types. It supports generalized combinatorial auctions with new algorithms for winner determination. It also allows bidding via graphically drawn price-quantity graphs. It has an expert system for helping the user decide which auction type to use. Finally, it supports mobile software agents that bid optimally on the user's behalf based on game theoretic analyses.

The second component, *eCommitter*, is a leveled commitment contract optimizer. In automated negotiation systems consisting of self-interested agents, contracts have traditionally been binding. Leveled commitment contracts—i.e. contracts where each party can decommit by paying a predetermined penalty—were recently shown to improve Pareto efficiency even if agents rationally decommit in Nash equilibrium using inflated thresholds on how good their outside offers must be before they decommit. *eCommitter* solves the Nash equilibrium thresholds. Furthermore, it optimizes the contract price and decommitment penalties themselves.

1 Introduction

Electronic commerce is taking off rapidly, but the full power of AI, algorithmic support, and game theoretic tools has not been harnessed to improve its efficiency. This paper presents *eMediator*, a next generation electronic commerce server that demonstrates ways in which these techniques can improve ecommerce both in terms of processes and outcomes. The result of our 2-year implementation effort is now available for use on the web at `http://ecommerce.cs.wustl.edu/emediator/`, see also (Sandholm 1999b). The components of *eMediator* include an auction house and a leveled commitment contract optimizer.

2 *eAuctionHouse*: A configurable auction

Several successful Internet auction sites exist such as eBay and OnSale. Our motivation to develop an auction server, *eAuctionHouse*, was to prototype next generation features, and to test their feasibility computationally and in terms of user comfort. *eAuctionHouse* allows users across the Internet to buy and sell goods as well as to set up auctions. It is a third party site, so both sellers and buyers can trust that it executes the auction protocols as stated. To our knowledge, it is the first—and currently only—Internet auction that supports combinatorial auctions, bidding via graphically drawn price-quantity graphs, and by mobile agents.

*This material is based upon work supported by the National Science Foundation under CAREER Award IRI-9703122, Grant IRI-9610122, and Grant IIS-9800994.

2.1 Combinatorial auctions

Usually, items are auctioned independently. If a bidder has preferences over combinations of items (as is common e.g. in electricity markets, equities trading, bandwidth auctions, and transportation exchanges (Sandholm 1993)), bidding is difficult. To determine how to bid on an item, the bidder needs to speculate which other items she will receive in other auctions. This introduces counterspeculation cost, and often leads to inefficient allocations where bidders fail to get the combinations they want and get ones they do not.

Combinatorial auctions can be used to overcome the need for lookahead and the inefficiencies, see e.g. (Sandholm 1993). In a combinatorial auction, bidders may place bids on combinations of items. This allows the bidders to express complementarities between items instead of having to speculate into an item's valuation the impact of possibly getting other items. *eAuctionHouse* supports a variety of combinatorial auctions:

OR-bids In the most commonly discussed combinatorial auction protocol, each bidder can bid on combinations of items, and her bids are joined with OR, i.e. any number of her bids can be accepted. Determining the winners so as to maximize the auctioneer's revenue is $\mathcal{NP}$-complete and inapproximable. *eAuctionHouse* tackles this via an optimal search algorithm that capitalizes on the fact that the space of bids is necessarily sparsely populated in practice (Sandholm 1999a).

XOR-bids OR-bids are based on the common assumption that the bids are superadditive: $b_i(S \cup S') \geq b_i(S) + b_i(S')$. But what happens if agent 1 bids $b_1(\{1\}) = 5$, $b_1(\{2\}) = 4$, and $b_1(\{1,2\}) = 7$, and there are no other bidders? The auctioneer could allocate items 1 and 2 to agent 1 separately, and that agent's bid for the combination would value at $5+4=9$ instead of 7. So, OR-bids focus on situations where combinational bids are introduced to capture positive complementarities among items. In many real world settings local subadditivities can occur as well. To address this, the bidders in *eAuctionHouse* can submit *XOR-bids*, i.e. bids on combinations such that only one of the combinations can get accepted. This allows the bidders to express general preferences with both positive and negative complementarities.

OR-XOR-bids While XOR-bids allow the bidder to express general preferences, in the worst case this would involve placing a bid for each of the $2^{\#items} - 1$ possible combinations. *eAuctionHouse* allows the user to submit multiple XOR-bids. We do this in table form, where each row is a combinational bid, and the rows are combined with XOR. These multiple bids are combined together with a non-exclusive OR. We do this by allowing the user to submit multiple tables. This method maintains full expressive capability, is a more natural way to input preferences, and leads to shorter input descriptions than XOR-bids only.

Other generalizations of combinatorial auctions *eAuctionHouse* also allows combinatorial double auctions, and combinatorial auctions where the agents can bid for multiple units of each item in a combination.

2.2 Bidding via price-quantity graphs

Price-quantity graphs are supported so bidders can express continuous preferences. E.g. when bidding for a larger quantity, one might only accept a lower unit price.

2.3 Support for choosing an auction type

eAuctionHouse supports a wide variety of auction types and helps the user choose an appropriate one. First, only auction types that are sensible based on game theoretic analyses or economics experiments are offered. Second, there is an expert system that restricts the choice of auction types given the auction setting. For any given auction setting, it tells the user what bid types should be acceptable, what price determination schemes should be chosen among, etc. The auction setting differs based on whether it is a single or double auction, whether there is one or multiple items, and whether there is one or multiple units of each item. Furthermore, the units can be divisible or indivisible. The bid types include a regular price bid, a price-quantity graph bid, an OR-bid, and an OR-XOR-bid. The pricing schemes include first-price, Vickrey, multi-unit Vickrey, Groves, and middle-price. The auctions in *eAuctionHouse* also have several other parameters.

2.4 *NOMAD*: Mobile agents in auctions

Our auction house supports mobile agents so that a user can have her agent actively participating in the auction while she is disconnected. Mobile agents that execute on the agent dock which is on (or near) the host machine of the auction server also reduce the network latency—a key advantage in time-critical bidding. Our auction server uses the Concordia agent dock to provide mobile agents a safe execution platform from where they can monitor the auctions, bid, set up auctions, move to other hosts, etc. The user has the full flexibility of Java programming at her disposal when designing her agent. We also provide an HTML interface for non-programmers where the user can specify what she wants her agent to do, and the system automatically generates the Java code for the corresponding mobile agent, and launches it. Some of these predesigned agents are alerting tools, others bid optimally on the user's behalf based on game theoretic analyses. This helps put novice bidders on an equal footing with experts.

3 *eCommitter*

Normal full commitment contracts are unable to take advantage of the possibilities that future events provide. Although a contract may be profitable to an agent when viewed *ex ante*, it need not be profitable when viewed after some future events have occurred. Similarly, a contract may have too low expected payoff, but in some realizations of the future events it may be desirable.

Leveled commitment contracts are a method for capitalizing on uncertain future events. Instead of conditioning the contract on future events, the level of commitment is controlled by decommitment penalties, one for each agent. If an agent wants to decommit—i.e. to be freed from the contract obligations—it can do so by paying the penalty to the other party.

Rational agents are reluctant in decommitting because there is a chance that the other party will decommit, in which case the former agent gets freed from the contract, does not have to pay a penalty, and collects a penalty from the breacher. (Sandholm & Lesser 1996) showed that despite such insincere decommitting, leveled commitment increases each contract party's expected payoff, and enables contracts in settings where no full commitment contract is beneficial to all parties. Given a contract and the probability distributions of the agent's outside offers (or analogously of the agents' future valuations of the contract), our *eCommitter* system computes the Nash equilibria, i.e. decommitting threshold pairs such that neither agent wants to change its decommitting strategy given that the other party does not change. The algorithm is conceptually highly nontrivial, but fast (Sandholm, Sikka, & Norden 1999). *eCommitter* also computes each agent's decommitting probability and expected payoff.

Furthermore, given no contract, but only the distributions of outside offers, *eCommitter* computes the welfare maximizing contracts (price and penalties). It gives a range of optimal contracts that are individually rational for both parties, and suggests the fair contract.

References

Sandholm, T., and Lesser, V. 1996. Advantages of a leveled commitment contracting protocol. In *AAAI*, 126–133.

Sandholm, T.; Sikka, S.; and Norden, S. 1999. Algorithms for optimizing leveled commitment contracts. In *IJCAI*. Extended version: Washington University, Dept. of Computer Science WUCS-99-02.

Sandholm, T. 1993. An implementation of the contract net protocol based on marginal cost calculations. In *AAAI*, 256–262.

Sandholm, T. 1999a. An algorithm for optimal winner determination in combinatorial auctions. In *IJCAI*. Extended version: Washington University, Dept. of Computer Science WUCS-99-01.

Sandholm, T. 1999b. *eMediator*: A next generation electronic commerce server. WUCS-99-02, Washington University, Dept. of Computer Science.