MoCAAS: Auction Agent System Using a Collaborative Mobile Agent in Electronic Commerce

Kwang-Yong Lee[a], Jung-Sup Yoon[b], Geun-Sik Jo[c]

[abc]Inha University Intelligent E-Commerce Systems Lab

253 Yong-Hyun-Dong, Nam-Ku, Inchon 402-751, Korea

Tel: +82-32-875-5863, E-mail: [a]antifire@eslab.inha.ac.kr, [b]jsyoon@eslab.inha.ac.kr, [c]gsjo@inha.ac.kr

Abstract

To get the items that a buyer wants in Internet auction, he must search for the items through several auction sites. When the bidding starts, he (the buyer) needs to connect to these auction sites frequently so that he can monitor the bid states and re-bid. A reserve-price auction reduces the number of connections, but this limits the user's bidding strategy. Another problem is equity between the buyer and the seller. Both the buyer and the seller should profit together within proper limits. In this paper, we propose an auction agent system using a collaborative mobile agent and a brokering mechanism called MoCAAS (Mobile Collaborative Auction Agent System), which mediates between the buyer and the seller and executes bidding asynchronously and autonomously. This reduces connection costs, offers more intelligent bidding, and solves the equity problem.

Keywords:

Auction; Brokering; Collaborative Mobile Agent

1 Introduction

Though EC's technology has grown steadily, it is still difficult to implement the negotiation between a buyer and a seller online. The Internet auction has been widely expanded as an alternative solution [1]. However, to search for items, monitor bid states, and re-bid, users need to connect to the auction site frequently. Also, first-time buyers do not know bidding strategies nor an item's value; therefore a competitor may cheat them, and they may lose the chance to buy items at a cheaper price. As a result, they may not buy the items that they want, or they may pay too much.

If the above problems are solved, the Internet auction will become one of the generalized EC markets. Thus, in this paper, we propose an auction agent system called MoCAAS (Mobile Collaborative Auction Agent System), which mediates between the buyer and the seller and executes bidding autonomously for the buyer. When a buyer submits a reserve-price and the identity of an item, the agent searches for the item among registered auctions. It then recommends auctions to the buyer and informs the buyer of the expected price for the item. When the buyer selects the best among the recommended auctions, the agent executes bidding for the buyer.

This section of our paper presents an overview of the MoCAAS system and its benefits. Section 2 presents an overview of the auction, the auction agent, and the collaborative mobile agent. Section 3 presents the architecture and workflow of MoCAAS. Section 4

presents the bidding processes of MoCAAS. Section 5. presents the brokering algorithm of MoCAAS. Section 6 reports experimental evaluation results. Finally, Section 7 briefly presents a summary and future considerations.

2 Related Works

2.1 The English Auction

The "auction" is the buying and selling of property through public bidding. The "English auction" is the most common and simplest type of auction [2]. Sotheby's and Christie's use this method for auctioning fine art. This is the method used at most Internet auction sites. In the English auction, the auction house will take bids in ascending order (a bidder must bid more than the "going price"). The highest bidder receives the item and the highest bidder pays for the item. The English auction is called "open" because every bidder knows all of the other bids and "ascending" because each bid must be higher than the one before. The method works with both *Single Unit* and *Multiple Unit* auctions. For example, if a case of coffee mugs is being auctioned and the bidders have to bid on the whole case, then this would be a single unit auction. If bidders can buy individual coffee mugs, then this is a multiple unit auction. In this paper, we set the English single auction as the target.

2.2 Other Auction Agent Systems

Several auction sites have solutions that offer a convenience to the users. eBay uses a reserve-price auction method. This allows the user to enter a reserve-price. As long as the auction is open and the user's reserve-price has not been reached, the agent bids the minimum amount necessary to become the highest bidder. However, this limits the user's choice of bidding strategy and may involve taking into account the effect of the "winner's curse". The winner's curse is the difference between the amount the winner paid and the next lower bid [2]. If the bidder bids the perceived valuation of the item and wins, the bidder will know that he paid too much because others valued the item less. To solve this problem, the agent allows the user to coordinate bids across multiple auctions automatically and select a bidding strategy [3].

Nomad [3] and Magnet [5] are auction agents using mobile agent mechanism. A mobile agent has the unique ability to transport itself from one system to another. This ability allows mobile agents to execute asynchronously and autonomously. Also, the mobile agents communicate with one another. Because of this ability, the auction agent using a mobile agent is smarter than the reserve auction method. Thus, the agents track bids in multiple auction houses in order to look for the best deal and/or coordinate the user's bids in the different auctions. However, these systems are inadequate when brokering between buyers and multiple auction sites. The brokering can reduce network traffic and increases efficiency.

3 MoCAAS Architecture & Workflow

MoCAAS consists of five main components:

- **A buyer-agent**
- **A broker-agent**
- **A bid-agent**
- **An auctioneer-agent**

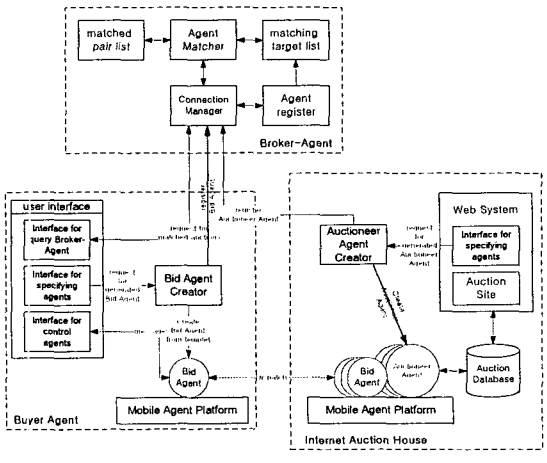

Figure 1: MoCAAS architecture

The architecture of a MoCAAS is illustrated in Figure 1. The buyer-agent offers a buyer interfaces for querying the broker-agent, specifying the bid-agents, controlling the bid-agents. An Interface for querying the broker-agent shows recommended auctioneer list and expected price of the item. An interface for specifying agents sends A bid-agent creator the information for creating the bid-agent. (Bidding strategy type, bidding schedule, etc.) Then, the bid-agent creator creates the bid-agent from template and registers the bid-agent with the broker-agent. An interface for control agents allows the user communicates with the bid-agent and controls the bid-agent's behaviors.

The broker-agent matches the buyers and the auctioneers, and informs buyers of the expected price of the item. A connection manager forwards messages to a agent register or a agent matcher. If the message comes from the buyer-agent and the message is query for recommending the auctioneer-agent, the connection manager forwards the message to the agent matcher. If the message comes from the buyer-agent or the Internet auction house and the message is request for registering the agent, the connection manager forwards the message to the agent register. The agent register registers agents with matching target list. The agent matcher matches a buyer-agent and an auctioneer-agent. It inserts matched pair into the matched pair list. The matched pair list haves buyer-seller pairs and expected price values.

The Internet auction house is the place that auctions are processed. A interface for specifying agents sends request for generated auctioneer agent to the auctioneer agent creator. The auctioneer agent creator creates the auctioneer agent from template and registers the auctioneer agent with the broker-agent. The created auctioneer-agent waits for the bid-agent in a mobile agent platform.

These agents collaborate with each other in order to execute an auction. When a new auctioneer-agent is created, it registers itself with a broker-agent. (1) The buyer submits the item's identity and reserve-price to the buyer-agent. (2) The buyer-agent submits the data received to the broker-agent. (3) The broker-agent searches for a recommendable auctioneer-agent and computes the expected price, then returns the results to the buyer-agent. (4) The buyer selects an auctioneer-agent among the auctioneer-agents recommended by the broker-agent. (5) The buyer-agent creates the bid-agent, and then dispatches it to the selected auctioneer-agent. (6) The dispatched bid-agent registers itself with the auctioneer-agent, reads the auction information from the auctioneer-agent's blackboard, and executes the bidding autonomously. (7) When the auction is started, the auctioneer-agent reads or writes the database at the auction site that it is present in. At this point, the auction site becomes synchronous with the auctioneer-agent. (8) The buyer-agent receives the intermediate auction information from the bid-agent, and controls the bid-agent. (9) When the auction ends, the auctioneer-agent un-registers itself with the broker-agent, and the bid-agent sends a "win" or "fail" message to the buyer-agent. (10) The winning buyer-agent exchanges transaction information in order to transact. (11) Figure 2 below presents whole MoCAAS workflow.

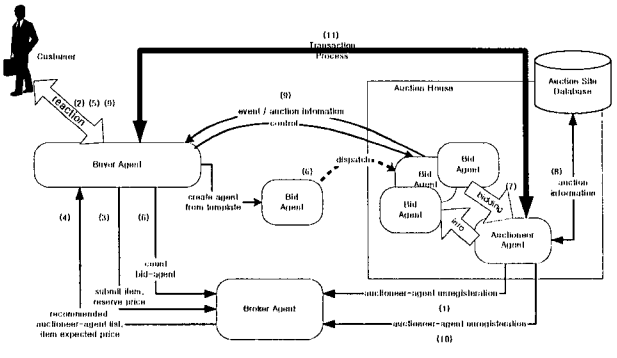

Figure 2: MoCAAS Workflow

As can be seen from the workflow, the buyer needs to connect to network once in order to start the bidding. From that point on, the buyer-agent executes the rest autonomously. Finally, the buyer-agent notifies the buyer of the bidding results.

4 Brokering Algorithm

Brokering between multiple buyers and sellers is similar to a continuous double auction (CDA). Therefore, MoCAAS's brokering algorithm is similar to CDA's. AuctionBot implements CDA with Mth-price and (M+1)st-price Rules [7]. In a set of L Single-unit bids, M are sell offers and the renaming N = L – M are buy offers. The Mth-price auction clearing rule sets the price at the Mth highest among all L bids. Similarly, the (M+1)st price rule chooses the price of the (M+1)st bid. Mth price is undefined if there are no sellers, and the (M+1)st price is undefined if there are no buyers [8].

MoCAAS's broker-agent uses these rules in order to recommend auctioneer-agents and determine the expected price. For this example, see the next list. This is the output of the broker-agent when making a match.

agentType: buyer
agentID: buyer1
ownerID: antifire
itemID: item1
reservePrice: 2400

expectedPrice: 2400

agentType: buyer
agentID: buyer2
ownerID: zzang
itemID: item1
reservePrice: 2300

expectedPrice: 2400

agentType: buyer
agentID: buyer3
ownerID: sstown
itemID: item1
reservePrice: 2600

expectedPrice: 2600

agentType: seller
agentID: seller1
ownerID: jang24
itemID: item1
initialPrice: 2500

matching: seller(seller1) sell 1 item(item1)
buyer(buyer3)
expectedPrice: 2600

agentType: seller
agentID: seller2
ownerID: kim77
itemID: item1
initialPrice: 2200

matching: seller(seller2) sell 1 item(item1)
buyer(buyer1)
expectedPrice: 2400

First, new bids (buyers or sellers) register steadily with the broker-agent. When buyers and sellers both are present in the broker-agent, the seller (seller1) matches the buyer (buyer3) by the Mth price rule. The broker-agent recommends a seller (seller1) for a buyer (buyer3), and informs the expected buyer (buyer3) of the expected price (2600). The matched pair is deleted from the matching target list and inserted matched pair list. The system continues to make matches. Broker-agent sends average of expected price about an item to buyer-agent.

Figure 3 shows that MoCAAS's broker-agent makes matches.

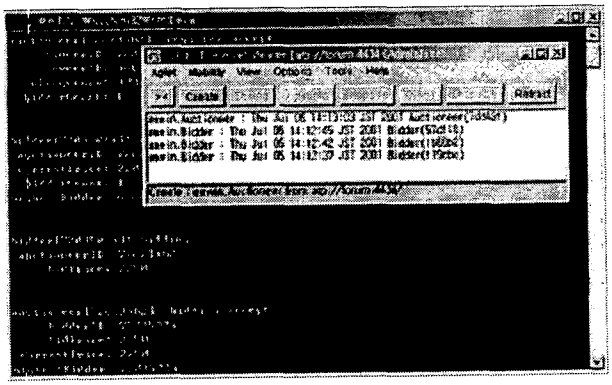

Figure 3: MoCCAS broker-agent

5 Bidding Processes

The dispatched bid-agent executes the bidding autonomously by exchanging messages between the bid-agent and the auctioneer agent. (See Figure4) First, when the bid-agent arrives at an auction host, the bid-agent registers itself with an auctioneer-agent in the auction host. (1→2) If the bid-agent is valid, the auctioneer-agent accepts the bid-agent and updates the information of accepted bidders. (2→3) When the bidder submits a bid, (3→4) the auctioneer-agent evaluates the bid. If the bid is valid, the auctioneer-agent accepts it and updates the current bid price. (4→3) When the auction ends, the auctioneer-agent informs all registered bid-agents of the auction results

and exchanges transaction information with the winning bid-agent. (4→5)

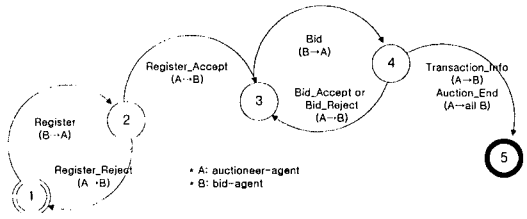

Figure 4: MoCAAS Bidding protocol

The next list is the message output between the bid-agent and the auctioneer-agent in the auction host.

```
from: bidder[72d3502a]
to: auctioneer[72c23ab2]
subject: register
ownerID: antifire

from: auctioneer[72c23ab2]
to: bidder[72d3502a]
subject: register_accept
ownerID: antifire
mFrequency: 30
bidderCount: 1

from: bidder[72d3502a]
to: debugging ouput
subject: read_blackboard
currentPrice: 2200
bidderCount: 1
highestBidder: none

from: bidder[72d3502a]
to: auctioneer[72c23ab2]
subject: bidding
bidPrice: 2250

from: auctioneer[72c23ab2]
to: bidder[72d3502a]
subject: bid_accept
current bidPrice: 2250
highestBidder: 72d3502a
```

The above list shows that the bid-agent registers itself with the auctioneer-agent, and submits a bid. "mFrequency" is the number of times the intermediate auction information is monitored per hour. The auctioneer-agent keeps the intermediate auction information on its blackboard [6]. The blackboard is opened to bid-agents. A bid-agent monitors the blackboard in "mFrequncy". "BidderCount" is the number of registered bid-agents.

6 Experimental Evaluation

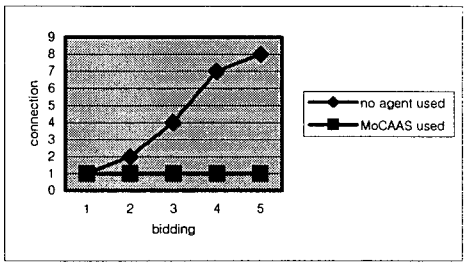

Figure 5: Number of connection

We selected an arbitrary item and compared MoCAAS's performance with the performance of existing auction sites. First, we compared an MoCAAS-used auction site with a normal auction site (no agent-used auction site) for number of connections in bidding. (See Figure 5) In the normal auction site, whenever the user bid, he connected with the site. There were also cases where there were connections for monitoring the auction events. For example, the user connected 4 times for bidding and connected 3 times for monitoring the auction states. However, when MoCAAS was used, the bid agent executed bidding processes automatically and remotely. This freed the user from having to poll the auction site repeatedly.

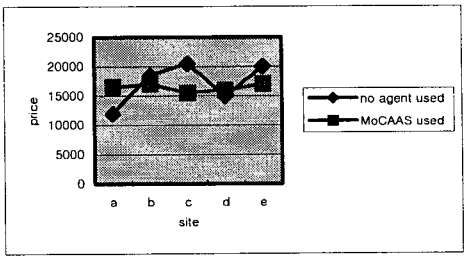

Figure 6: Price distribution

Second, we compared MoCAAS with normal auction sites for price distribution. (See Figure 6) As can be seen, the difference in price for the same item between the normal auction sites was more than between the MoCAAS-used auction sites. Because the normal auction sites do not support a price-comparison

service between multiple sites, the first-time buyers sometimes experienced the winner's curse. However, the MoCAAS-used sites support a price comparison service between multiple sites. This informs the user of the expected price of the item. The buyers buy the same item at a similar price. These experiments show that MoCAAS reduces the number of connections with the auction sites and is an efficient method for pricing an item.

7 Conclusion

MoCAAS is an auction agent, which helps buyers by supporting autonomous bidding and recommending auctions. Because it adopts a mobile agent mechanism, it reduces the number of connections. In this paper, the broker-agent plays the role of implementing the coordination of bids across multiple auction sites before dispatching the bid-agent to the auction site. However, after dispatching the bid-agent to the auction site, does not allow the broker-agent to do so. In future studies, we will employ a buyer-agent that sends multiple bid-agents to multiple auction sites to buy an item and coordinate the bid-agents in order to buy the item at the lowest price. To do this, we will use the heuristic algorithm that predicts the increment of bid price in the auction.

In the wireless Internet environment, communication costs rise in proportion to a rise in the number of connections. The effect is more than that in the wire Internet environment. Therefore, using MoCAAS in the wireless Internet environment result in a greater profit than in the wire Internet environment. To achieve this, we propose to deploy part of MoCAAS into wireless devices or make new components so that MoCAAS can connect with wireless devices.

8 Acknowledgement

This work was supported by a grant No. 2000-1-30300-016-3 from Korea Science & Engineering Foundation.

References

[1] Jae Kyu Lee, Hyng Rim Choi, Hyun Soo Kim, Kyoung Jun Lee, "Principles of Electronic Commerce", Bupyoungsa, 1999.

[2] Auctus development, inc. (http://www.auctusdev.com/)

[3] Tuomas Sandholm, Qianbo Huai, "Nomad: Mobile-agent System for an Internet-Based Auction House", IEEE Internet Computing, pages 80-86, 2000.

[4] Danny B. Lange, Mitsuru Oshima, "Programming and Deploying Java Mobile Agents with Aglet", Addison-wesley, 1998.

[5] Erik Steinmetz, John Collins, Scott Jamison, Rashmi Sundarewara, Bamshad Mobasher, Maria Gini, "Bid Evaluation and Selection in the MAGNET Automated Contracting System", selected paper of First International Workshop on Agent Mediated Electronic Trading AMET-98, pages 105-125, 1998.

[6] Gerhard Weiss, "Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence", the MIT Press, 1999.

[7] Peter R. Wurman, Michael P. Wellman, William E. Wash, "The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agent", Proc. Second International Conference on Autonomous Agents (Agent-98), 1998.

[8] Peter R. Wurman, William E. Walsh, and Michael P. Wellman. "Flexible double auctions for electronic commerce: Theory and implementation", Submitted for publication, 1998.