REAL-TIME MULTI-AUCTIONS AND THE AGENT SUPPORT

Hong Liu
Department of Electrical and Computer Engineering
University of Massachusetts Dartmouth
North Dartmouth, MA 02747-2300

Shouhong Wang
Department of Marketing/Business Information Systems
University of Massachusetts Dartmouth
North Dartmouth, MA 02747-2300
swang@umassd.edu

Fei Teng
Department of Electrical and Computer Engineering
University of Massachusetts Dartmouth
North Dartmouth, MA 02747-2300

ABSTRACT

Electronic auctions have become popular. Recently, multicast technology is changing the Internet environment for e-commerce, and will make available more forms of auctions than were ever practical in the past. This study proposes an online real-time multi-auctions model. The prerequisite of the implementation of this new auction model is the integration of the multicast technology and software agent technology. In this paper, the architecture of the agents that support activities leading to the effectiveness of online real-time multi-auctions is examined. A prototype of the agent, which is built with Java in the multicast environment, is further discussed.

1. Introduction

Auctions have been an important type of business to sell merchandise based upon effective pricing methods [Hanson 2000]. Conventionally, auctions are commonly applied to sell unique and unusual items including celebrities' personal property and art. Since the Internet became the e-commerce media, online auctions are virtually applied to all kinds of commodities ranging from low-price CD's to expensive real estate [eShop 2000].

Compared with face-to-face selling and negotiation methods, the auction is certainly efficient. The philosophy behind the auction is to let the market determine the true value for the product. Bidders feel that they correctly evaluate the items they bid on, and auctioneers feel that they receive the highest winning bids for the items they sell.

However, the online auctions currently available on the Internet are non-real-time. Non-real-time auctions on the Internet have many limitations. From the auctioneers' point of view, a non-real-time auction has a long business cycle time, and might be risky. From the bidders' perspective, a non-real-time auction does not allow a bidder to respond quickly to market dynamics. Due to these limitations, the current form of online auctions is not fully equivalent to conventional face-to-face auctions.

This study proposes a real-time multi-auctions (RTMA) model for unsealed auctions. Multi-auctions take advantage of multicast, a new communication technology on the Internet, and call for the support of software agents. In the following sections, we define the RTMA model, explain the special communication technology requirements, analyze the structure of software agents for RTMA, and discuss the implementation of RTMA.

2. Real-Time Multi-Auctions

2.1 Conventional Online Uni-Auctions

Although there are a variety of auction types (e.g., English auctions, Dutch auctions, etc.), the conventional auction can be modeled as "one auctioneer to many bidders," as shown in Figure 1.

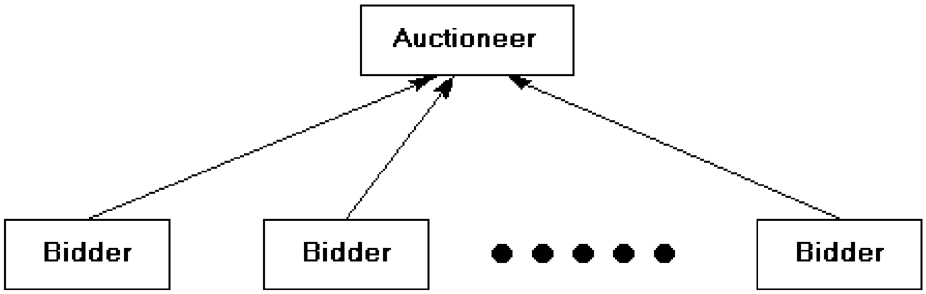

Figure 1. The Conventional Uni-Auction Model

At present, all online auctions are non-real-time uni-auctions. They are so-called non-real-time because auction information circulated on the Internet can has perceivable delays. They are also called uni-auctions because these auctions are auctioneer-centered, and bidders are not supposed to be searched by auctioneers. This non-real-time uni-auction form has many drawbacks. Since the conventional online uni-auctions are not real-time, the auction cycle time is usually long. In many cases, bidders have enough time to cooperate and reach agreements not to outbid each other. This phenomenon of collusion in auctions has the overall effect of lowering the winning bids [Cramton 1998; Kelly and Steinberg 2000]. Because of the information delay, information from one auction cannot easily be utilized by a bidder in another auction. Also, online auctions of this form do not differ from the nature of face-to-face auctions and are still "one auctioneer to many bidders."

2.2 Real-Time Multi-Auctions

Recently, although many online auction sites (e.g., [Amazon 2000; eBay 2000; AuctionWatch 2000]) provide bidders with the archive of past auctions and even allow bidders to manage multiple auctions, these online auctions are non-real-time in nature, since no-one can control delays of auction information circulating on the Internet. To compete with face-to-fact auctions, some auction sites, such as [eauctionroom 2000], allow bidders to participate in "live" auctions. However, without the support of multicast techniques, these "live" auctions would delay the bidding process significantly especially when the number of bidders increases, as we have already experienced.

Two correlated critical issues must be resolved in order to make online auctions more effective. First, since the time factor is so crucial for the majority of types of auctions, the present mass communication environment is no longer adequate for online auctions. Accordingly, we would improve the current online auctions by further exploiting information technology to create a real-time environment. Second, in the real-time auction environment, bidders would be interested in participating different auction markets of correlated commodities simultaneously to make "globally optimal" bidding. Therefore, we must further exploit the advantages of the Internet to make it possible for bidders to conduct bidder-centered online auctions. As our discussion of agent support later in this paper shows, RTMA are not only possible, but also feasible. RTMA are different from conventional non-real-time uni-auctions in three major aspects.

(1) Like face-to-face auctions but unlike online uni-auctions, any bidder in a RTMA can receive bidding information in real-time fashion.

(2) Unlike face-to-face auctions or uni-auctions, any RTMA bidder could join multi-auctions and utilize real-time auction information from multiple auctions to make better bidding decisions. This feature of RTMA allows the bidder to coordinate actions across different auctions on the same or correlated commodities (e.g., similar or same cameras, or computers and compatible peripheral equipment).

(3) Like face-to-face auctions but unlike online uni-auctions, a RTMA can be conducted within a short time duration (e.g., several hours).

Current online auctions offer a variety of bidding types. However, the RTMA model is proposed only for unsealed auctions. Under RTMA, auctioneers can organize auctioneer-centered auctions, and bidders are able to carry out bidder-centered auctions. In this sense, the RTMA model is a new model for auctions.

The proposed RTMA model is depicted in Figure 2.

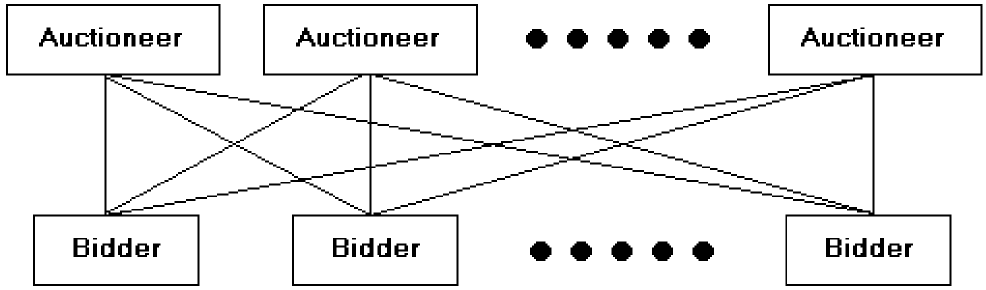

Figure 2. The Multi-Auctions Model

Apparently, RTMA makes auctions more effective and efficient, and provides more power to the auction business. However, to fully implement RTMA, one must acquire data communication capacity for "real-time" and information processing capacity for "multi-auctions." In the following section, we discuss the two premises for RTMA.

2.2. Multicast Technology

Information traffic in traditional IP networks is based on the unicast mode; that is, one sender to one receiver (point-to-point) data communication. Unicast results in a severe waste of network bandwidth and often causes network congestion or even failure of information delivery. In the online auction case, as the number of participants and the number of items in auctions grows, the requirement for delivering real time pricing and bidding information will generate a huge amount of traffic on the network. As a result, unpredictable delays of data transmission across the Internet occur in on-line auctions. Furthermore, since unicast requires a particular data connection for any individual client request, it imposes a heavy burden on the auctioneer's server. This not only makes real-time transactions impossible, but also requires more investments in computing resources for auctioneers.

Multicast technology [Comer 1999] enables one copy of digital information to be received by multiple receivers. The different communication forms of unicast and multicast are depicted in Figure 3.

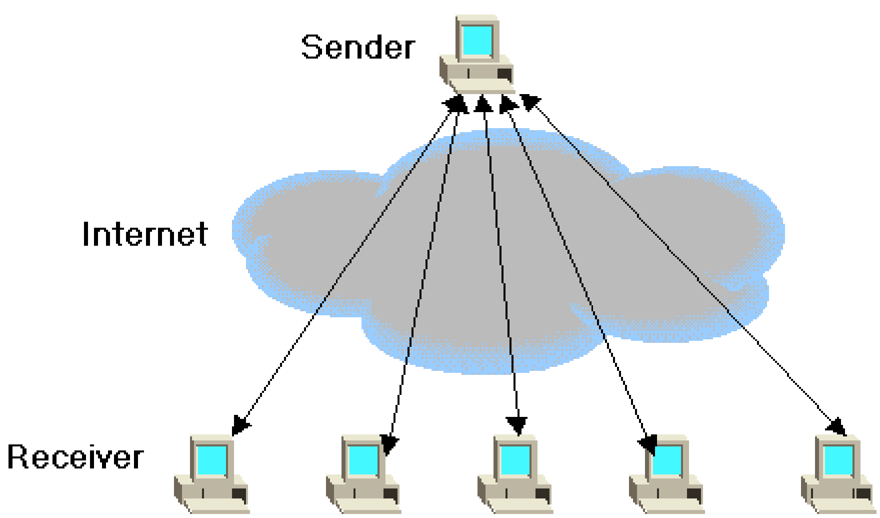

Figure 3(a). Unicast – Traffic Overload on the Internet

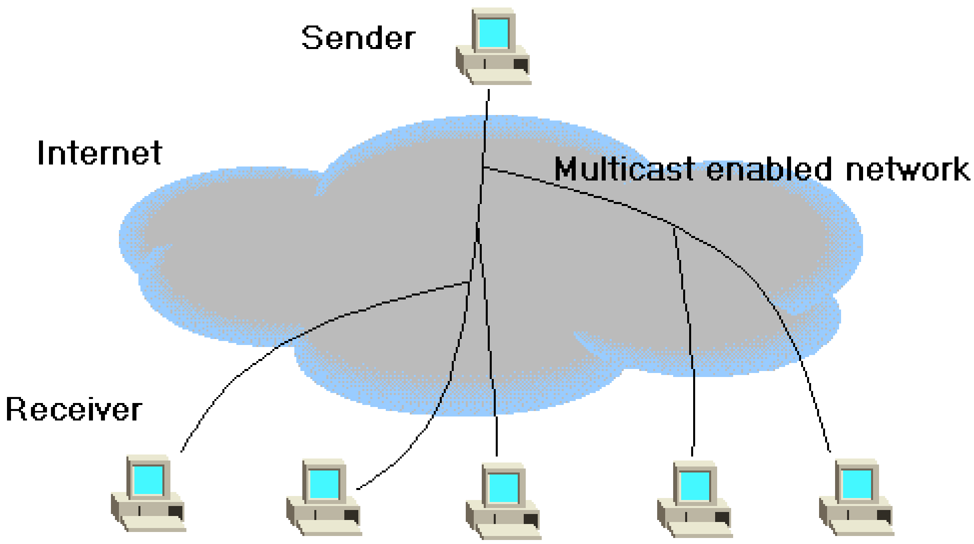

Figure 3(b). Multicast – Reduced Traffic on the Internet

In multicast, users (computers) form groups. Each multicast group is assigned a unique class D address. On the Internet, the IP addresses ranging from 224.0.0.0 through 239.255.255.255, called class D addresses, are reserved for multicast usage. When a computer wants to launch a multicast session, it sends its packets to a particular class D address, and any host in this multicast group will receive the packets. This schema allows physically spreading computers to become members of the same group. The membership of a group is dynamic; that is, any computer on the Internet is able to join or leave a multicast group at any time. The number of group members is virtually unlimited, and the boundary of a multicast session is specified by the Time to Live (TTL) parameter of the multicast datagram.

A protocol called Internet Group Management Protocol (IGMP) for multicast has been developed [Deering 1988]. This protocol is used between a host and the immediate routers that are connected to this host. Coordination between the multicast routers is defined by different multicast routing protocols. Many protocols are currently available, including DVMRP (Distance Vector Multicast Routing Protocol) [Deering 1988], PIM-DM (Protocol Independent Multicast-Dense Mode), and PIM-SM (Protocol Independent Multicast-Sparse Mode) [Deering *et al.* 1996]. Each of these has its advantages in particular cases depending upon the density of group members, communication distance, and dynamics of grouping.

For RTMA, in the multicast setting, the auctioneer and bidders can form a group. Any member of the group can be a sender and send a single copy of information for multicast to all other members. All auction transactions can be real-time.

2.3. Agents

In the RTMA environment, the major burden of information processing is located on the bidder's side, as the bidder needs to review all auction sessions in the real-time mode and make decisions as quickly as possible. To relieve bidders from intensive monitoring and timely decision making, one must use software agents.

Agents technology is a step towards the solution in the rapid changing business environment of the e-commerce era [Wang 1999]. A software agent is a software entity that functions continuously and autonomously in a particular environment [Shoham 1997]. Software agents carry out activities in the manner of human agents; they can be used to solve or alleviate the problems in e-commerce. Generic intelligent search engines and browsers, learning agents, and knowledge sharing agents have begun to appear in the software market. To survive in the e-commerce environment, people are increasingly turning to advanced software agent techniques. In fact, artificial intelligence, with its roots in machine learning, is now experiencing a renaissance as new tools emerge to make e-commerce more tractable [O'Leary 1997].

The agent-oriented approach [Shoham 1997] has proposed a particular the set of activities necessary to create software agents. However, the agent-oriented approach has not demonstrated its ability to generate all kinds of agents, especially for the e-commerce field. Currently implementation of the agent-oriented approach still relies on traditional software development tools such as object-oriented languages (e.g., [Bigus & Bigus 1998]). In fact, the agent-oriented approach can be thought of as a specialization of the object-oriented approach [Bradshaw 1997].

From this perspective, an agent is a set of objects that are integrated to achieve common goals by accomplishing tasks and coordinating activities. Hence, objects are the elementary units in the construction of agents. In the next section, we will examine the architecture of agents for RTMA by identifying these elementary objects.

3. Structure of Real-Time Multi-Auctions Agents

An agent is a super object, which has its identification and specific operations, necessary to accomplish its mission. An agent object for RTMA incorporates these fundamental elements.

1. Goal. Agents possess abilities such as autonomy, collaborative behavior, and inferential capability [Etzioni & Weld 1995]. A common characteristic of these abilities is that an agent is goal-directed. Agents can initiate actions on their own without the user's intervention and explicit instruction for the actions. Hence, the core element of an agent is the goal descriptions for the agent.

2. Operational Task. An agent can be employed to perform a basic operational task(s). One of the most popular approaches to task description is hierarchical task analysis [Annett & Duncan 1967]. Using hierarchical task analysis, a bidding task is formally described as a hierarchical structure.

3. Cognizance. A software agent can demonstrate intelligent behavior if it possesses knowledge, or cognizance concepts [Riecken 1994]. Theoretically, a knowledge base is a set of cognizance objects representing production rules, semantic networks, cognitive maps, or other forms of knowledge [Bates 1994]. For example, a RTMA agent remembers all rules commonly used by bidders.

4. Mnemonic instrument. One of the missions of software agents in e-commerce is to reduce information overload [Maes 1994; O'Leary 1997]. An intelligent agent needs several types of specific mnemonic functions, such as information filtering [Maes 1994], knowledge acquisition [Elofson & Konsynski, 1991], and case matching [Kolodner & Mark 1992].

5. Decision Instrument. Effective e-commerce must include decision support functions [Hinkkanen *et al*. 1996; Huh 1993]. A type of genuine elementary model that can be used for all kinds of decision support systems is called a decision instrument. In RTMA, a decision instrument of the agent for effective bidding is the combination of accounting models and statistical models.

6. Interface. Interface descriptions specify the dialogue between the user and agents. There is a large literature on the elemental descriptions of interface design in the human-computer interaction field (e.g., [Baecker & Buxton 1987, Shneiderman 1987]). These principles are applicable to the present context for the user-agents interface.

In summary, to take part in RTMA, the bidder must consider employing software agents. Generally, a RTMA agent consists of goal, operational task, cognizance, mnemonic instrument, decision instrument, and interface objects, as summarized in Figure 4.

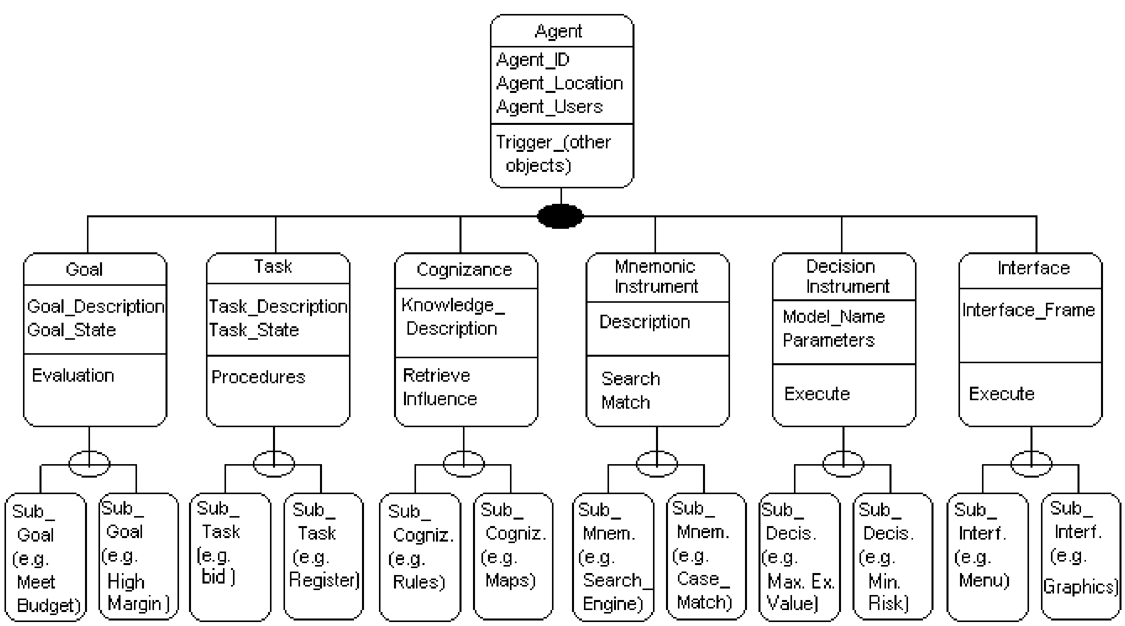

Figure 4. The Structure of RTMA Agents

4. Implementation

To demonstrate the effectiveness of the RTMA model, we implemented a prototype of the RTMA system. This implementation included two phases. In the first phase, the multicast environment was set for RTMA. In the second phase, a prototype of the software agent for RTMA was built.

In setting the multicast environment for RTMA, a local area network was employed. The topology of the test bed for the multicast environment is depicted in Figure 5.

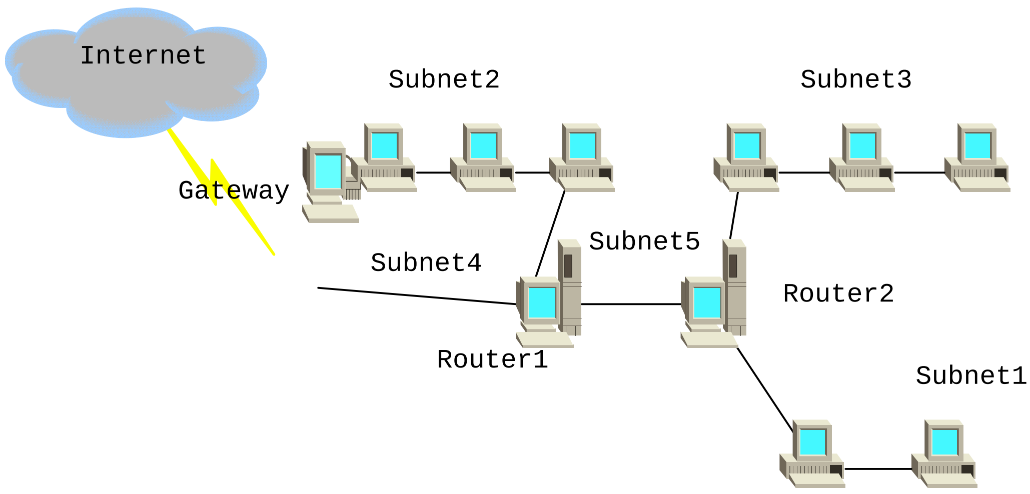

Figure 5. Topology of the Test Bed

As shown in Figure 5, the test bed consisted of five sub-nets, which were separated by two routers. All the computers were connected with each other using 10Base2 (thin Ethernet). The test bed was connected to the Internet through a gateway computer running the Linux operating system (Redhat 6.2). The gateway computer performed the Network Address Translation (NAT) [Tsirtsis 2000] for all the hosts inside the test bed. This packet translation gave these computers direct Internet access, yet did not require additional configuration on the computers inside the test bed. The two routers were Pentium PCs with multiple network-interface cards (NIC), and Linux operating systems (Debian distribution). Other hosts in the test bed were all Pentium PCs with different operating systems including Windows NT, Solaris (i86), and Linux. This setting allowed us to observe the effects of the software agents on different platforms.

The multicast routing schema was achieved by the cooperation of the operating systems on the routers, and a multicast routing daemon program. We reconfigured and recompiled the Linux operating system kernel (version 2.2.14), enabled the multicast routing function of the OS, and then installed a daemon program called mrouted on the routers. The mrouted program implemented the multicast routing algorithm and executed the multicast policies. It instructed the OS kernel to direct multicast datagrams and perform the routing job.

In this test, we also employed a tool called MGEN, a shareware originally developed as a part of the Real-Time Information Transfer and Networking (RITN) project by the Naval Research Laboratory (NRL) and Defense Advanced Research Projects Network (DARPA) [Clark 1986]. We used it to generate multicast as well as unicast background traffic to simulate the real world situation on the Internet.

In the second phase of the prototype RTMA system implementation, we developed software called NiuBid, using Java (JDK1.2.2) as the major developing tool. Logically, NiuBid included two parts, representing the auctioneers and bidders respectively. We used Oracle8i for Solaris as the supporting database management system for the auctioneers' side. The database manipulations in this system were implemented using SQLJ, a convenient Java plugin language designed by Oracle that provides SQL functions within Java. The SQLJ source code was first translated into Java code using the SQLJ translator. The Java code was then compiled into Java byte code using the Java compiler. On the auctioneer’s side, a Java program was running to manipulate all the operations on the database and transmit bidding information of the auction to the logical multicast group address. On the bidder's side, a Java application that implemented the agent ran on each bidder's computer. The Java application had multiple threads so that the agent was able to monitor multiple auctions concurrently and receive all bidding information circulated within the multicast group.

Directed by the goal object, the agent was able to adjust its bidding policies during the bidding process according the rules or parameters previously set by the user of the agent. In designing the agent, we programmed various methods in the goal description to enable the agent to achieve the optimal result for the bidder. These methods were in turn to trigger a variety of objects, as discussed below.

Using cognizance objects, the agent retained the objects of current available auctioneers and auction products. The agent was able to apply different identifications or credit cards for individual auctioneers. It was also able to memorize important facts, such as the time left for each of the auctions, and significant events, such as outbids.

Through the use of task objects, the agent was able to automatically perform various bidding tasks. Upon launching a new auction, the NiuBid agent started to watch the bids. For the multi-auction purpose, the agent was able to find the same type of the product, like cameras and computers, from multiple auctioneers' sites, and keep monitoring the auction dynamics on these auctioneers' spots. Technically, the agent generated a special thread to retrieve the time information from each of the auctioneers, and thus kept itself synchronized with all other agents in bidding. When a new offer in the auction came out, the agent was able to retain this information in real-time manner and trigger a bidding response.

The major contribution of this research study is the integration of the agent technology and multicast technology. Accordingly, the most important feature of NiuBid was to coordinate different auctions, and make bidding decisions autonomously or with minimum human interference. Several general principles were set for the agents' behavior. For the bidder, the timing of bid placement is important. Many on-line auction companies like Amazon offer first-bid discount for the earliest bidder. Therefore, the first principle of the bidder agent was to bid as early as possible. Second, the agent could bid as frequently as possible. The agent was able to keep a close watch on the auctions and respond with another highest offer to "intimidate" other bidders, and therefore to lower the cost for winning the bid. Third, the agent was able to make snipes by bidding at the last minute of the auction.

In any cases, the user was allowed to set the ceiling for bidding, and the agent could never make an offer more than this amount. If the bid goes too high, the agent might simply drop out of the auction, and wait for the next similar item to appear on the Internet. One of the advantages of RTMA was that the agent was always able to make comparisons among several existing bids to make the best decision for the user.

Generally, a bidder can also become an auctioneer, or *vice versa*, in the RTMA model. In this case, the agent was able to help the user monitor different auction sites, and maximize the number of potential bidders. This would be especially useful for those people who make a living buying and selling goods through on-line auctions. The agent was able to decide when to initiate a new auction. If there were similar items being auctioned on other auction sites, the agent might decide not to open the new auction until those others closed.

Because of the features discussed above, the NiuBid agent was able to make bidding decisions. When outbid in all its active auctions, the agent would raise its bid by a given increment (e.g., $10 in the prototype) to take the lead. When the current offer exceeded the predetermined limit, the NiuBid agent would either stop bidding or alert of the user (bidder) by chiming and displaying a special icon on the screen. It was also able to give first priority to the user's urgent demand for a particular product.

The user (bidder) was able to control the agent and receive a report from the agent through the interface. The interface of the agent includes three major tabs (see Figure 6). The first tab was the auctioneer tab. It displayed a list of available auctioneers. The second tab allowed the bidder to set important parameters, such as the highest price that the bidder willing to pay and the number of auctions to monitor concurrently, to rule the behavior of the agent. The third tab provided information and images of the product, as well as other critical information about the auctions, such as the time left for bidding and the current best offer. Examples of the interface of the NiuBid agent are shown in Figure 6.

The term "real-time" means that the response of an application should be limited within a prescribed range of time, and the loss of precision should remain above an acceptable level. In this sense, the current online auctions are not real-time and must be improved. We have conducted simulation experiments on our test bed. The experiment results (available upon request) show that, on an average, the bandwidth required by an application with multicast can be one-twelfth of that required by one with unicast, while the data receiving rate in multicast can be twice faster. However, without agent support, RTMA are unrealistic because of the imbalance between the limited human information processing capacity and the fast multicast transactions. This study aims to integrate the agent technology and multicast technology for the new RTMA model. The prototype has demonstrated the effectiveness of the system in unsealed multi-auctions. Using this system, the bidder is able to receive needed real-time information. This in turn makes it possible for bidders to conduct bidder-centered auctions; that is, the bidder is able to deal with several auctions simultaneously. A software agent can be used to improve the multi-auction environment. The software agent is able to conduct routine transactions and goal driven decision making autonomously.

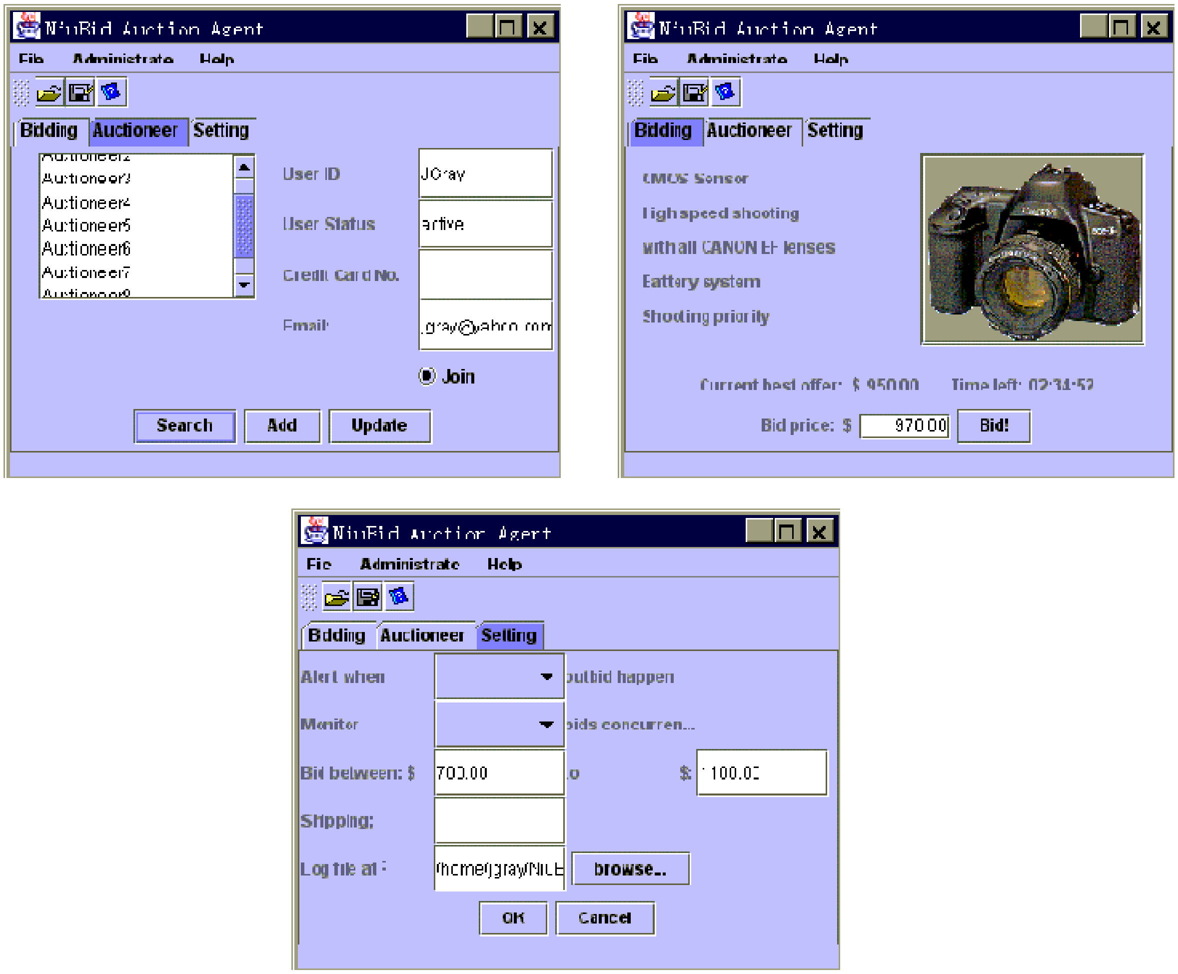

Figure 6 The NiuBid Agent

5. Conclusion

In this paper, we proposed the RTMA model for online auctions, and explained the advantages of RTMA over conventional online auctions. We discussed multicast and software agent, the two key technologies for implementing the RTMA model, and their integration. Finally, we demonstrated the design of the software prototype of RTMA. Laboratory experiments with the prototype have indicated that RTMA are feasible and very promising in e-commerce.

Further research is needed to investigate the comprehensive structure of objects for RTMA software agents related to authentication, security, and intelligent bidding decision making functions.

Acknowledgement: The comments of three anonymous reviewers have contributed significantly to the revision of the paper. Special thanks to Professor Louis Habicht who helped the authors improve the literal presentation. This research study was funded by the University of Massachusetts Dartmouth Joseph P. Healey Endowment Award 2000.

REFERENCES

Agorics, Collusion and Tricks, 1996, <http://www.agorics.com/auctions/auction11.htm> [Accessed May 10, 2000].
Amazon <http://www.amazon.com> [Accessed September 1, 2000] 2000.
Annett, J. and Duncan, K. D., Task analysis and training design. *Occupational Psychology*, 41, 1967, 211-221.
AuctionWatch <http://www.auctionwatch.com> [Accessed September 1, 2000] 2000.

Baecker, R. M. and Buxton, W. A. S., *Readings in Human-Computer Interaction: A Multidisciplinary Approach*, Morgan Kaufmann, San Mateo, CA, 1987, 427-437.

Bates, J., The role of emotion in believable agents. *Communications of the ACM*, 37(7), 1994, 122-125.

Bigus, J. P. and Bigus, J., *Constructing Intelligent Agents with Java*, John Wiley & Sons, NY, 1998.

Bradshaw, J. M., An introduction to software agents. In J. M. Bradshaw (Ed.) *Software Agents*, AAAI Press, Menlo Park, CA, 1997, 3-46.

Clark, D. D., The design philosophy of the DARPA internet protocols, *The Proceedings of SIGCOMM*, *Computer Communication Review*, 18(4), 1988, 106-114.

Comer, D. E., *Internetworking With TCP/IP*, 3rd ed. Volume 1, Prentice Hall, Upper Saddle River, NJ, 1999, 289-302.

Cramton, P., Ascending auctions, *European Economic Review*, 42(3-5), 1998, 745-756.

Deering, S., Multicast routing in Internetworks and extended LANs, *The Proceedings of SIGCOMM*, *Computer Communication Review*, 25(1), 1995, 89-101.

Deering S., D. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu and L. Wei. The PIM architecture for wide-area multicast routing. *IEEE/ACM Transactions on Networking*, 4(2), 1996, 153-162.

eauctionroom <http://wwww.eauctionroom.com> [Accessed September 1, 2000] 2000.

eBay <http://www.ebay.com> [Accessed September 1, 2000] 2000.

eShop, Auctions, 2000, <http://auctions.msn.com/Scripts/
glossary.asp> [Accessed May 10, 2000].

Elofson, G. and Konsynski, B., Delegation technologies: environmental scanning with intelligent agents. *Journal of Management Information Systems*, 8(1), 1991, 37-61.

Etzioni, O. and Weld, D. S., Intelligent agents on the Internet: Fact, fiction, and forecast. *IEEE Expert*, 10(4), 1995, 44-49.

Hanson, W., *Internet Marketing*, South-Western College Publishing, Cincinnati, OH, 2000.

Hinkkanen, A., Kalakota, R., Saengcharoenrat, P., Stallaert, J., and Whinston, A. B., Distributed decision support systems for real time supply chain management using agent technologies, 1996, <http://ecworld.utexas.edu/ejou/articles/art_1.html> [Accessed January 30, 1998].

Kelly, F. and Steinberg, R. A combinational auction with multiple winners for universal service, *Management Science*, 46(4), 2000, 586-596.

Kolodner, J. and Mark, W., Case-based reasoning. *IEEE Expert*, 7(5), 1992, 5-6.

Maes, P., Agents that reduce work and information overload. *Communications of the ACM*, 37(7), 1994, 31-40.

O'Leary, D. E., The Internet, Intranets, and the AI renaissance. *(IEEE) Computer*, 30(1), 1997, 71-78.

Riecken, D., M: An architecture of integrated agents. *Communications of the ACM*, 37(7), 1994, 107-116.

Shneiderman, B., *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley, Reading, MA, 1987.

Shoham, Y., *An overview of agent-oriented programming.* In J. M. Bradshaw (Ed.) *Software Agents*, AAAI Press, Menlo Park, CA, 1997, 271-290.

Tsirtsis, G., Network Address Translation - Protocol Translation (NAT-PT) RFC2766, February 9, 2000, <http://www.faqs.org/rfcs/rfc2766.html> [Accessed May 30, 2000].

Wang, S. *Analyzing Business Information Systems: An Object-Oriented Approach*, CRC Press, Boca Raton, FL, 1999.