A Framework for Bidder Agents in Multiple Simultaneous Auctions

Paulo André Lima de Castro

Technological Institute of Aeronautics – ITA

São José dos Campos , SP, 12228-900, Brazil

+55 12 39475981

pauloac@ita.br

Jaime Simão Sichman[1]

Intelligent Techniques Laboratory

University of São Paulo, Brazil

Av. Prof. Luciano Gualberto, 158 – trav. 3

05508-900 São Paulo SP BRAZIL

+55 11 30915397

jaime.sichman@poli.usp.br

Abstract

Among hundreds of auction sites in the internet, some offer software agents that search auctions, monitor them and/or submit bids in these auctions. However, most of these agents deal with just one auction, instead of acting in multiple simultaneous auctions. The framework proposed in this paper facilitates the implementation of bidder agents, especially to act in multiple simultaneous auctions, by solving the problem of search and monitor auctions and a high level interface with the auction server.

1. INTRODUCTION

There are many auction sites in the internet with several auction schemes. However, the use of software agents to automate the tasks involved in act in auctions is very new. There are few auctions systems based in software agents and even less systems with bidder agents, which can monitor and bid in multiple simultaneous auctions [9].

1 Partially supported by CNPq, Brazil, grant numbers 304605/2004-2 and 482019-2004-2.

Very interesting research in auctions agents are made under the Trading Agent Competition [4]. It's an open competition for agent mediated electronic commerce research groups. The competition participants have to develop travel agents whose task is to organize itineraries for a group of clients. There are three types of travels goods: flight tickets, hotel room reservations and entertainment tickets. The agent's objective is to secure the goods necessary to satisfy the particular desires of its clients and do to so as inexpensively as possible. The travel goods are traded through auctions whose types are derived from classic types presented in this paper and the travel goods are strongly related (for example, the days of flight tickets define in which days the agent should have hotel room reservations). The work we present in this paper is different from the works made for TAC, because TAC approaches only superficially the problem of trading commodities products that can be bought from several sources. The main motivation of our work is to develop a framework for bidder agents that take advantage from the possibility of buying a product from any of several possible sellers.

Among several auctions systems, we can find some very innovative academic auctions house based in agents, such eMediator [9], AuctionBot [5] and BiddingBot [7]. We will describe these systems in section 3. However, most of these systems do not support agents to monitor and bid in multiple simultaneous auctions. This feature is very important, because to monitor and bid in many auctions can help the user to obtain better economic results due to the increase in business options. On the other hand, increasing the number of auctions may turn impossible for a human being to monitor all the auctions. In this situation, the use of software agents can be very interesting to the bidder.

This paper presents a framework to facilitate the building of bidder agents to multiple simultaneous auctions and some experimental results that we obtained using this framework. We call this framework of AAS (Auction Agent System).

2. WHY BIDDER AGENTS FOR MULITPLE AUCTIONS?

The use of agents to automate electronic commerce activities is an issue targeted by many researchers, especially in automation of auctions. However, most of the researchers focus in designing agents that act in only one auction. Despite the fact that this issue is very important; we believe that in the near future the demand for capable agents to act in many simultaneous auctions will increase very fast, because these agents can create more efficient markets [2].

There are several papers that relate research to obtain better algorithm to bid in auctions [1, 8, 10]. These works support the following auctions types: English, Dutch, First price sealed-bid (FPSB) and Vickrey's auction. However, they don't implement their strategies in real electronic auction servers. Anthony [1] presents a simulation of an environment with agents and some experimental results, but the paper advises that is just a simulation in a simplified environment and not an implementation for a real auction server. We intend to facilitate these implementations, developing a system that can wrap the complexity of each auction server API, and perform the search for new auctions and the monitoring of the chosen auctions. These tasks are fundamental for a bidder agent to work, but they are not dependent of agent's strategy.

3. AUCTION SERVERS FOR BIDDER AGENTS

This section presents a short comparative analysis among some selected auction systems [5, 7, 9], considering interesting features from the bidder's point of view for multiple simultaneous auctions. We don't intend to judge the quality of the analyzed systems; we just aim to choose the most appropriate system to use as an infrastructure for AAS. These systems were analyzed based on three specific features, which are described below. The systems features are shown in table 1.

- **Bidder Agents for Multiple Auctions**: this feature shows if the system provides bidder agents that can operate in multiple simultaneous auctions.
- **Auction Types Supported**: this feature describes which basic auction types (English, Dutch, First Price and Sealed Bid and Vickrey) are supported by the system.
- **Bidder System or Auction Server**: Some of these systems can act like a wrapper to third party auction sites. This feature points if the system acts like an auction server, as a wrapper or as both.

The eMediator [9] has shown some advantages over the others systems, because it provides support to third party agents and enables all the basic auction types. Even if the AuctionBot [5] also shows these features, eMediator has other additional features, like support for combinatorial auctions, mobile agents and others types of auctions beyond the basic auction types. Despite the fact that these additional features are not required to AAS, they can be very useful for a future development. For these reasons, we have chosen the eMediator as the auction server for AAS. Our proposal, named AAS+eMediator, is represented in the last column of table 1.

Table 1. Systems Features

System / Feature	BiddingBot [7]	AuctionBot [5]	EMediator [9]	AAS + eMediator
Bidder Agents for Multiple Auctions	No	No	No	Yes
Auction Types Supported	English	All	All	All
Bidder System or Auction Server	Bidder System	Auction Server	Both	Both

4. THE AUCTION AGENT SYSTEM (AAS)

In this section, we describe the architecture of the system, its communication process and the two bidding strategies used to implement AAS agents. These strategies were used to demonstrate the use of AAS, we don't argue that theses are the better possible strategies for bidding agents. In fact, they were chosen by their simplicity as we will show in this section.

4.1. Architecture

AAS (Auction Agent System) approaches two problems that are faced by the bidder agent's developer. These problems are: (i) low abstraction level in the auction server's API in all auction servers that we have studied and (ii) the lack of services to find and monitor target auctions. The eMediator API is based on formatted strings transmitted over TCP/IP protocol. The AAS provides a set of classes that wraps that API, thus offering a higher level of abstraction and an object oriented interface. The services of searching and monitoring are provided by the AAS agents Auction Searcher and Auction Monitor, respectively.

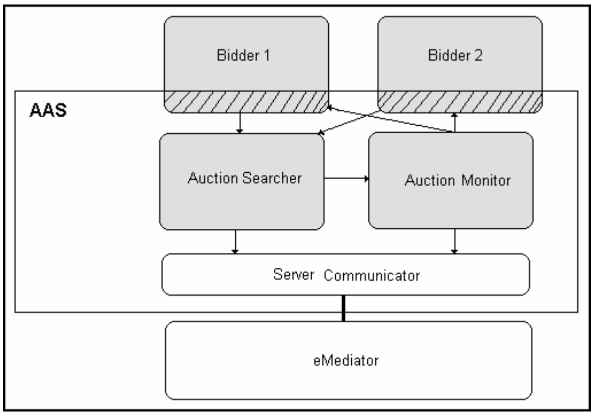

Figure 1 - AAS Software Architecture. The gray rectangles are AAS agents, the others are traditional software. The lines symbolize communications among agents. The Server Communicator establishes communication with the server, through the server's API (bold line).

The AAS architecture and its relationship with eMediator are presented in figure 1. The Auction Searcher agent and Auction Monitor agent provides the identification of auction which can be interesting to the bidder(s) agent(s) and monitor such auctions, respectively. The search service (Auction Searcher) finds the target auctions for the bidder agent, using the product name given by this agent. After finding the target auctions, he informs the Auction Monitor which auctions should be monitored and which bidder agent has asked for them. This job is continuous, i.e., the Auction Searcher keeps constantly looking for new target auctions. Meanwhile, the Auction Monitor gets information about all target auctions requested by Auction Searcher and sends them to the bidder agent.
In fact, the Auction Searcher and Auction Monitor may provide their services to several bidder agents and not only to two bidders as in the example of figure 1. Furthermore, the AAS may work with more than one auction server. In order to deal with another different auction server, we only need to implement a new Server Communicator (SC) specific to deal with the API of the new auction server. The Bidder, Auction Searcher and Auction Monitor agents are thus capable to communicate with several different SCs.

4.2. Communication in AAS

The communication among agents is performed by an open source system called SACI (Simple Agent Communication Infrastructure) developed by Hubner and Sichman [6]. All messages are written in KQML and transported by SACI system. SACI allows that the agents may be located in different machines, facilitating several configurations. For example, different machines can be used to execute Auction Searcher, Auction Monitor and Server Communicator, as well as to execute each bidder's agents.

The communication with the user is performed through a graphic user interface. The user informs its preferences (product of interest and reserved price) to the bidder agent through a customized screen for each strategy. Each agent can request other information besides the identification of the product and its reserved price, according with its strategy. The monitoring of the agent's actions can be done by the user through a screen containing the summary of the main data related to the agent's performance, independently of the agent used.

4.3. Bidder Agent's Strategies in AAS

Two bidding agents with different strategies were implemented to demonstrate the use of AAS, and these agents were selected from articles on agents´ performance in multiple auctions [1]. We have opted for implementing two different strategies to facilitate the comparison among the results obtained by each one and to demonstrate the capacity of AAS to aid the development of more efficient strategies. The bidder agents developed are the: Greedy Agent and the RT Agent, which use respectively the strategies called Greedy and Remaining Time. Such strategies have the following characteristics:

- Greedy [3]: It is one of the simplest strategies for multiple auctions. The agent gives bid in the auction with the smallest current price (and randomly in case of tie) and with bid price just lightly above it. The agent continues to give bids until it holds a winning bid in some auction or until all the auction prices are higher than its reserved price. The agent doesn't submit new bids if it already holds a winning bid, in order to avoid having more than one winner bid in the group of auctions;
- Remaining Time [1]: This strategy is defined according to the remaining time to the end of each auction. As the end of the auction approaches, the agent increases the value of the bid. The choice of the auction to offer a bid is done through the calculation of the expected utility in each auction. In the case the agent decides that a bid should be offered, this offer will be sent to the auction with largest expected utility.

Although there are several other possible strategies, the ones described above were selected because of their simplicity. The Greedy strategy is a generalization for the case of multiple auctions of the dominant strategy for English auctions. On the other hand, the RT strategy presents, besides its simplicity, the capacity to be used in any of the four basic types of auctions mentioned in section 2.

5. RESULTS ANALYSIS

We have performed a series of experiments using the Greedy and RT bidder agents. In the execution of these experiments, the auction server we have used was the eMediator system [9]. We have also implemented another bidder agent that uses the dominant strategy for English auction, called MonoAuction Agent. Such agent can be described as a simplification of the Greedy strategy, for the case of a unique auction. In this work, the MonoAuction Agent was used to simulate the other possible bidders that operate in each one of the auctions, since such a situation is the most common in practice.
Due to the great conceptual differences between the two strategies, the agents' behaviors are also quite different. Greedy agent stays in the auction with lower value current and it almost always submits bid, meanwhile RT agent presents few bids (in this experiment, just two) and with values not directly related to the current value of the auction, since sometimes the bid’s value can be quite higher than the average values of the auctions. This happens when the time defined by the agent’s owner is close to the end. In this situation, the agent offers a bid whose value is close to the reserved price. A more detailed description of the quantitative results is out of scope of this paper, and may be found in [11].
We did not intend in this work to define the best strategies to act in multiple auctions or to determine which one of the implemented strategies is better. Instead, we intended to develop a system to simplify the agents' construction, which agents would be able to operate in the eMediator in multiple simultaneous auctions. Such objective has been reached, as the experiments described in this section demonstrate the operation of AAS bidder agents that implement two different strategies (Greedy and RT). They also demonstrate that the search agents' service (Auction Searcher) and monitoring services (Auction Monitor) work aptly, as well the communication interface with the server of auctions eMediator.
On the other hand, observing the small size of necessary code to implement agents Greedy and RT (114 and 268 lines of code Java, respectively) and the fact that they don’t have to control the search and the monitoring of the auctions (done by agents supplied by AAS), we can conclude that AAS performs the role of facilitating the development of bidder agents capable to act in simultaneous multiple auctions. Furthermore, we still developed an agent (MonoAuction) that implements the dominant strategy for an English auction that facilitates the comparison experiments among bidder agents for multiple auctions and bidder agents for only one auction.

6. CONCLUSIONS

With the use of AAS, the development and evaluation of new strategies for multiple simultaneous auctions will be facilitated, because AAS provides built-in services for search and monitor auctions and a high level interface to work with an electronic auction server (eMediator). This interface (Server Communicator) could be extended to work with others auction servers. Therefore, the framework removes such work from the agents' developer, which can focus in his main problem: the development of better bidding strategies.

7. REFERENCES

[1] Anthony, P.; Hall, W.; Dang, V. and Jennings, N.R. Autonomous agents for participating in multiple on-line auctions. In: IJCAI Workshop on E-business and the Intelligent Web, Seattle-WA, 2001. **Proceedings.** IJCAI, 2001.

[2] Preist, C.; Byde A.; Bartolini, C. Economic dynamics of agents in multiple auctions. In: 5th International Conference on Autonomous Agents. Montreal, 2001. **Proceedings.** Montreal, 2001. p.545-551.

[3] Byde, A. The dynamic programming model for algorithm design in simultaneous auctions. In: Workshop on And-Commerce (WELCOM ’01), 2. Heidelberg - Germany. **Proceedings.** Heidelberg: Springer-Verlag, 2001. p. 152-163.

[4] M. Wellman et al., “The 2001 Trading Agent Competition,” E*lectronic Market*s, vol. 13, no. 1, 2003, pp. 4–12.

[5] Wurman, P.; Wellman, M.; Walsh, W. The Michigan Internet AuctionBot: THE configurable auction server for human and software agent. In: International Conference on Autonomous Agents (AGENTS), 2. **Proceedings.** Minneapolis: St. Paul, 1998. p. 301-308.

[6] Hubner, J.; Sichman, J. SACI: Uma ferramenta para implementação e monitoração da comunicação entre agentes. In: Simpósio Brasileiro de Inteligência Artificial (SBIA), Atibaia-SP-Brasil, 2000. **Open Discussion Track**. Atibaia: Monard, M.C. e Sichman, J.S. (Eds.), 2000. p. 47-56.

[7] Ito, T.; Fukuta, T.; Sycara, K. BiddingBot: The multiagent support system for cooperative bidding in multiple auctions. In: Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS´2000), 4. Boston-USA, 2000. **Proceedings.** Boston, 2000, p. 399-400.

[8] Preist, C.; Bartolini, C.; Philips, I. Algorithm design for agents which participate in multiple simultaneous auctions. In: Agent Mediated Electronic Commerce, 3, Berlin, 2001. **Proceedings.** Berlin: Springer-Verlag, 2001. p. 139-154.

[9] Sandholm, T; eMediator: Next Generation Electronic Commerce Server. In: International Conference on Autonomous Agents (AGENTS), 4. Barcelona-Spain, 2000. **Proceedings.** Barcelona, 2000. p.341 - 348.

[10] Byde, A.; Preist, C.; Jennings, N.R. Decision Procedures for Multiple Auctions. In: Autonomous Agent Multi-Agent Systems (AAMAS), Bologna-Italy, 2002. Proceedings. Bologna: ACM Press, 2002. p.613-622, part 2.

[11] Castro, Paulo A.L. Uma infra-estrutura para agentes arrematantes em múltiplos leilões simultâneos. Master Thesis. Intelligent Techniques Laboratory. University of São Paulo. Brazil. 2003.