

Real-Time Trust Management in Agent Based Online Auction Systems*

Rinkesh Patel, Haiping Xu and Ankit Goel
Computer and Information Science Department
University of Massachusetts Dartmouth
North Dartmouth, MA 02747, USA
{g_rpatel, hxu, agoel}@umassd.edu

Abstract. *Agent based online auctions have not yet become popular because they are not trustable. One of the major concerns in agent based online auctions is the shilling behavior problem, which makes winners have to pay more than what they should pay for auctioned items. In this paper, we propose a real-time trust management module for agent based online auction systems using role-based access control mechanisms. As one of the key components of the trust management module, a security agent can actively monitor online auctions in order to detect abnormal bidding behaviors in real-time. To illustrate the feasibility of our approach, we implemented a prototype real-time trust management module for agent-based online auction systems, and demonstrated how shill agents could be efficiently detected.*

1. Introduction

One of the most popular electronic commerce activities in recent years has been the use of online auction systems. Among the various auction types, the English auction has emerged as the preferred form for online auction systems (e.g., eBay) due to its characteristics of multiple bids and ascending bidding price [1, 2]. As the number of users and products increases, more time is required for a user to search and bid for an auctioned item. To cope with this problem, agent based online markets have come into play. An agent based online auction system is a multi-agent system [3] that comprises software agents to handle tedious tasks on behalf of human users. Each agent is autonomous and capable of taking actions to fulfill its goal. Thus, in an agent based online auction system, an agent can represent a user to search and bid for a product based on the constraints defined by the user.

However, with the rapid rise in the number of users, fraudulent behaviors in online auctions become more and more severe. The British Sunday Times recently revealed that shill biddings were very common on eBay [4]. A shill bidding is an act of bidding against other bidders in order to raise the auction price, so a winner has to pay more than

what he should pay for an auctioned item [2]. In a trustworthy online auction system, buyers must trust sellers to provide the services they advertise, and not indulge in shill bidding; while sellers must trust buyers to be capable of paying for goods or services, and be authorized to make purchases on behalf of an organization. Trust in the sellers' competence and honesty will influence a buyer's decision on choosing sellers. In addition, users also must trust an auction house for not disclosing their personal information. Thus, there is a pressing need for a trust management system to maintain trust among users as well as with the online auction system.

In this paper, we propose a real-time trust management model to establish trust for agent based online auction systems. In our proposed model, a security agent is responsible for keeping track of each transaction and detecting unusual activities, such as shill biddings; while an authorization module can update a user's role and access permissions dynamically. Due to real-time actions against any abnormal auction activities, our trust management model can effectively maintain trust for agent based online auction systems.

The rest of this paper is organized as follows. Section 2 discusses about related work. Section 3 describes agent based online auction systems. Section 4 introduces a real-time trust management module integrated with a security agent. Section 5 presents an example to show how shill agents can be detected in real-time. Section 6 provides conclusions and our future work.

2. Related Work

There are two main strands of work to which our research is related, i.e., work on agent-based online auction system and work on trust management in e-commerce. Ito and his colleagues proposed *BiddingBot* as a multi-agent system that supports co-operative bidding [5]. In their approach, bidding decisions are actually made by users rather agents. Ogston and Vassiliadis proposed a peer-to-peer agent-based auction system for continuous double auctions [6]. They found that peer-to-peer auctions are able to display price convergence behavior similar to that of centralized auctions. In Collins and his colleagues' work, a multi-agent system for contract negotiation was

* This material is based upon work supported by the Chancellor's Research Fund and UMass Joseph P. Healey Endowment Grants.

presented [7]. The system can be used as a testbed for online auctions; however, it may have problems with secrecy of bids, non-repudiation, and manipulation of bids. Although the above efforts are useful in justifying the feasibility of agent-based approach for online auctions, there are no attempts so far to provide security mechanisms to prevent an agent-based online auction system from being abused. Therefore, it is still hard to convince users to adopt the existing agent-based approaches for practical usage.

On the other hand, most of the previous work related to trust management in e-commerce tried to secure online transactions, and establish trust among users by proposing different trust models [8, 9, 10]. Trust management using reputation models are based on a user's prior history and feedback from other users. For example, the reputation based trust model used by eBay has a very simple rating scheme for users. As one of the major drawbacks of this approach, it is possible for a user to provide counterfeit ratings for other users with a dummy account. Zacharia and Maes implemented a social mechanism of reputation management in Kasbah, in which a central system keeps track of users' explicit ratings, and uses these ratings to compute a person's overall reputation in a directed graph [10]. However, it is not clear how the agents may collect the ratings in an open agent-based environment.

Our work is closely related to a trust management model proposed by Herzberg and his colleagues [11], which was later extended by Mouri and his colleagues for consideration of changes in user's internal state [12]. In their proposed models, a trust management system consists of a trust establishment module and a role-based access control (RBAC) module [13]. However, their models are either "stateless" in nature, or use state information only when a user starts a new session. Thus, their approaches can not ensure trust among users in real-time.

In this paper, we propose a real-time trust management model for agent based online auction systems. Our proposed model can be used to establish and maintain trust among agents based on both agents' history information and real-time state information. To monitor and detect any undesired behaviors such as shilling behaviors in an agent based online auction system, a security agent is designed and implemented. In addition, we isolate various security related policies in different modules, so the policies can be updated dynamically.

3. Agent Based Online Auction System

An agent based online auction system is a multi-agent system that facilitates online auction activities on behalf of human users to make users' life much easier. We have developed a prototype agent based online auction system using the JADE agent development framework [14]. Figure 1 shows a client-server architecture of our agent based online auction system, which consists of various types of software agents, such as search agent, bidding

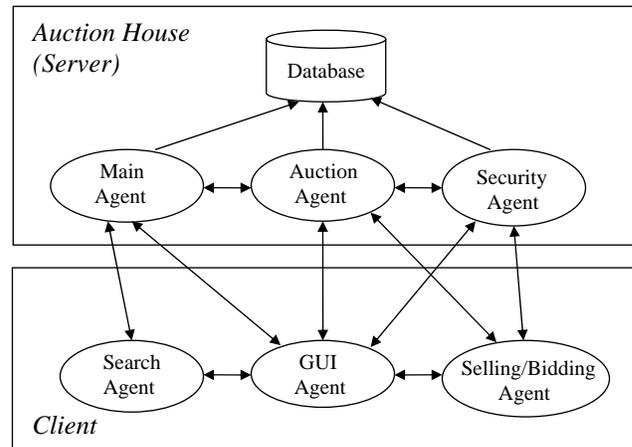


Figure 1. Architecture of agent based auction systems

agent, and auction agent. In particular, a security agent is introduced to provide security mechanisms for detection of undesired bidding behaviors.

The agent based online auction system is managed by an auction house administrator and used by various sellers and buyers. The auction house is implemented at the sever side with three major types of agents, namely the main agent, the auction agent, and the security agent. The main agent works as a controller for the auction house, and is responsible for creating new accounts for users, creating auction agents, and also responding to queries for items or auctions from agents at the client side. For each new auction, a corresponding auction agent is created to handle its auction related activities such as posting bids. While an auction is running, an agent representing a user can put bids on auctioned items; meanwhile, the corresponding auction agent is responsible for updating bidding activities for all involved agents. At the end of an auction, the auction agent notifies the winner of the auction, and passes the control back to the main agent. As a major component for security, the security agent monitors all online auction transactions performed by bidding agents.

The agents that work on behalf of human users are implemented at the client side, which involves three major types of agents, namely the search agent, the selling/bidding agent, and the GUI agent. A GUI agent receives commands from a user, and updates the user interface when messages are sent and received. A search agent can automatically search and join an auction on behalf of a user. Finally, a selling/bidding agent is responsible for initiating auctions or automatically placing bids on behalf of a user according to user defined bidding strategies. Note that a user can be a seller and a bidder at the same time.

In the agent based online auction system, a user can configure a bidding agent by providing auction related information, such as the type of items they are interested in, maximum value for that item, and bidding strategies for how to put bids during an auction. A configured bidding agent will run autonomously, and make decisions on behalf of the user during the bidding process.

4. Trust Management for Online Auction Systems

4.1 Shilling Behaviors

A shill bidding is a deliberate activity of placing bids in order to artificially raise the price of an auctioned item. Although shilling behaviors are prohibited in most of the online auction houses, e.g., eBay, it is very easy for malicious users to disguise themselves and put shill bids.

As most of the auction houses allow users to create new accounts using false information, a seller can create a new dummy account and pretend to be a valid bidder to bid on his own auction for shilling purpose. A shill user may also get help from his friends, immediate employees, and relatives to put fake bids using their auction accounts. When normal buyers realize that they have to pay extra for an auctioned item due to shilling activities, the credibility of the online auction house will surely be affected. To maintain trust among users as well as with the auction house, it is necessary to provide security mechanisms to detect shilling behaviors in real-time, and restricts further abnormal activities done by shill bidders.

Shilling behaviors could be much more severe in an agent based online auction systems because detection of shill bidders can be more difficult than in ordinary online auction systems, where auction activities are continuously monitored by human users. Furthermore, shill bidders may take advantages of the agent technology to introduce more shilling activities that are hard to detect. The major goal of this paper is to propose a real-time trust management module that can detect shilling behaviors and takes appropriate actions accordingly in a timely manner for agent based online auction systems.

4.2 An Overview

Figure 2 is an overview of our proposed trust management module in an agent based online auction system. From the figure, we can see that a human user can configure an agent to initiate an auction as a seller or put bids on an auctioned item as a buyer. Before an agent starts to work, it must go through a trust management module for security purpose. The agent needs to send a

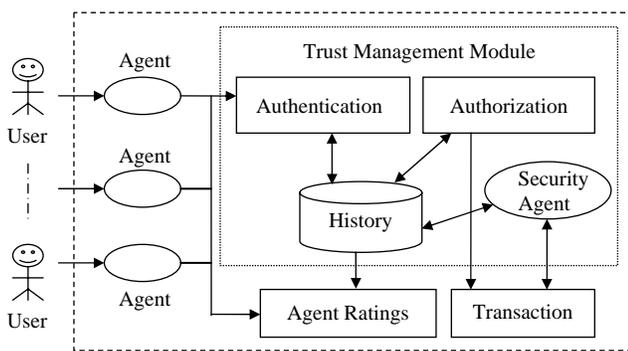


Figure 2. Trust management module

digital certificate or user credentials to the trust management module for authentication and authorization. Once the user configured agent is authenticated and authorized, it will be allowed to place requests for auction related activities. During the auction process, a configured agent can check current status or ratings of other configured agents in order to make proper decisions on choosing the right auction. Meanwhile, a security agent is designed for monitoring auction transactions for any suspicious bidding behaviors.

4.3 Trust Management Module

The trust management module (TMM) defined in Figure 2 is a key component in an agent based online auction system for trust maintenance, which can be further refined as shown in Figure 3. From the figure, we can see that the trust management module consists of a number of sub-modules such as authentication, authorization, state and history modules. As one of the major features of our TMM module, the security agent works closely with other modules of the TMM to maintain trust among agents in real-time. The authorization module, the access control module, and the security agent have their own policy rules defined by the auction administrator. Each set of policy rules are modularized in a corresponding database that can be updated dynamically without shutting down the agent-based online auction system.

Both the history module and the state module are parts of the TMM that are used to store and maintain the activities performed by user configured agents. When a user configured agent provides its digital certificate to the authentication module, the authentication module checks the certificate against previously stored information in the history module. If the authentication process is passed, the agent receives its initial pass, and is ready to make requests to perform auction activities. However, to make a request, the agent must also go through the authorization module, which consists of two major procedures, namely the role assignment and the access control. The role assignment process assigns a role to the configured agent dynamically by applying role assignment policies, called *RA Policies* based on gathered information related to the corresponding user. The access control process grants or restricts the access to auction related activities for the user configured agent based on access control policies, called *AC Policies*. The access control mechanism also determines how frequently the security agent should monitor a configured agent's auction transaction activities. After being authorized, the configured agent can start to make requests for auction related activities with certain permissions. Meanwhile, the security agent continuously monitors auction related activities in the auction system according to security agent policies called *SA Policies*. Once the security agent detects any shilling behaviors, the security agent determines the severe level of the shilling behaviors, and updates the current state information of the

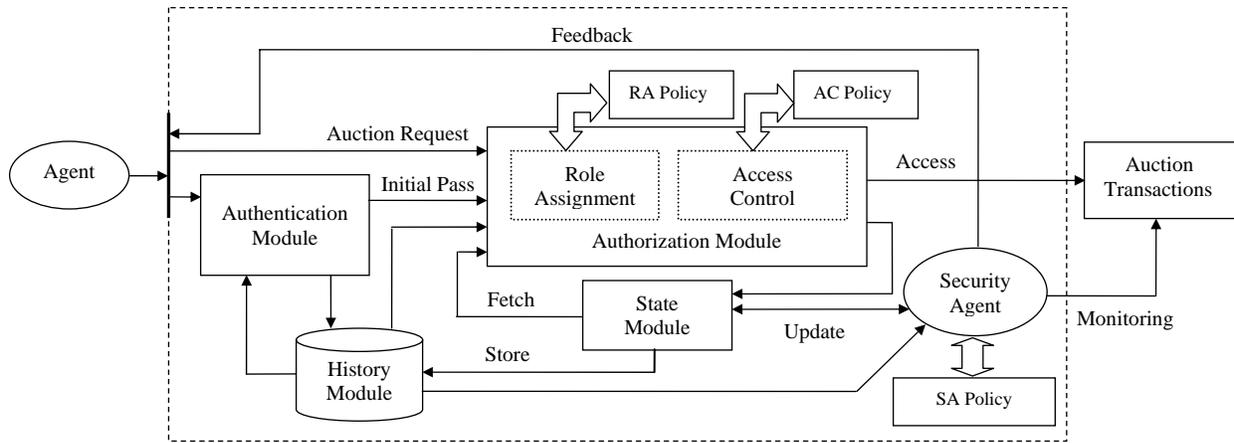


Figure 3. Refinement of the trust management module

skill bidder. Furthermore, the security agent notifies all participating configured agents about the shilling behavior of the skill bidder in the corresponding auction.

4.4 History Module and State Module

The history module stores information about users' previous auction activities over a certain period of time. Examples of such information include previously assigned roles, access information, shilling behaviors, and feedback information. After each successful transaction of a configured agent, the information in the history module is updated, and is ready to be accessed by the security agent and the trust management module for decision making.

The state module stores information related to the configured agents and their current activities, which includes currently assigned agent roles, granted resource access information, and possible shilling behaviors. The state module information is used along with the history module information to determine a configured agent's next dynamic role assignment by the role assignment module.

The information stored in the state module can be updated by both of the security agent and the authorization module. Current state information of the configured agent is the vital information used in a role assignment process for an agent's next bidding activity. After each successful transaction, information stored in the state module is saved into the history module for future use.

4.5 Authorization Module

In our proposed agent-based auction system, all requests made by an agent are controlled by the authorization module (Figure 3). In other words, in order to perform any auction related activities, an agent must first get an appropriate role and access permissions from the authorization module. We now describe in more details for the two major components in the authorization module, i.e., the role assignment module and the access control module, as follows.

Role Assignment Dynamic role assignment is performed according to predefined *RA Policies* stored in a role assignment database. The needed information for the computation includes the following: (1) the configured agent's history information, number of positive and negative feedbacks, and feedback status from the history module; (2) the user's current role and shilling behavior information from the state module. According to the *RA* policies, an agent can be assigned to one of the following five types of roles: *most trusted*, *trusted*, *average*, *untrusted*, and *most untrusted* for both sellers and buyers. As an example of role assignment rules, the following policy written in Prolog defines the conditions for assigning the *most trusted* buyer (*mtb*) role to an agent.

```
%If the current role is mtb or tb, and the
agent's reputation score is high enough.
conditions_for_mtb(HIST,CUR_ROL,SHILL_STATUS,POS_
FB,NEG_FB):- HIST>=0.8,
(is_identical_to(CUR_ROL,mtb);
is_identical_to(CUR_ROL,tb)),
(is_identical_to(SHILL_STATUS,clean);
is_identical_to(SHILL_STATUS,probable)),
POS_FB>=1000, NEG_FB=<(0.1*POS_FB).
```

According to the above rule, an *mtb* role is assigned to a bidding agent when the agent satisfies requirements such as having more than 1000 positive feedbacks, having less than 100 negative responses, not doing shilling in the last transaction, and taking a role of either *most trusted* buyer (*mtb*) or *trusted* buyer (*tb*) currently.

Access Control The access control module grants or denies an agent the access to resources requested by the agent. It may also restrict a bidding agent to perform certain auction activities for a period of time, if the agent has any shilling behaviors in its previous history.

A newly registered agent, which starts by getting a role of *average* buyer, is assumed to be trustable, so it shall have the privilege to perform auction activities. During the auction time, if an agent's role is downgraded (e.g., from a role of *average* buyer to a role of *untrusted*

buyer), it signifies that undesired activities have been done by the agent. In this case, the access control module may give warnings to the agent or restrict the agent to perform further activities for a certain period of time. If an agent is restricted to participate in any auction related activities for a certain period of time, the access control module sets the penalty status as *active* for the agent, and will deny all requests by that agent. The following is an example of *AC Policy* in Prolog that defines how different penalties can be applied and how different security levels will be set according to different situations of role changes.

```
% When a user's role has been downgraded
(is_identical_to(CUR_ROLE,tb),is_identical_to(LAS
T_ROLE,mtb))->
(penalty_assess(PENALTY,FIRST_TIME,oneday),assign
ed_value(SEC_STATUS,level3));
(is_identical_to(CUR_ROLE,avgb),is_identical_to(L
AST_ROLE,tb))->
(penalty_assess(PENALTY,FIRST_TIME,oneweek),assi
gned_value(SEC_STATUS,level2));
(is_identical_to(CUR_ROLE,ub),is_identical_to(LAS
T_ROLE,avgb))->
(penalty_assess(PENALTY,FIRST_TIME,twoweeks),assi
gned_value(SEC_STATUS,level2));
(is_identical_to(CUR_ROLE,mub),is_identical_to(LA
ST_ROLE,ub))->
(penalty_assess(PENALTY,FIRST_TIME,onemonth),assi
gned_value(SEC_STATUS,level1));
```

Note that the security level assigned (from 1 to 4, with level 1 being the highest security level) will be used by the security agent to determine the way the bidding agent should be monitored for abnormal behaviors when the bidding agent's auction activities are resumed.

A configured agent, whose role has been downgraded due to its past undesired behaviors, may gain back trust by refraining itself from performing undesirable activities after the restricted time period expires. When an agent has shown sufficient evidence for trustworthiness, the role assignment module may upgrade the agent's role according to predefined *RA Policies*. In addition, to prevent further undesired bidding behaviors, for those agents with high security level, the security agent will monitor them more closely and thoroughly for any activities performed by them when their bidding activities are resumed.

4.6 Security Agent and Detection Rules

To make online auction system trustworthy and to ensure the bidding process reliable, we should prevent and minimize undesired bidding behaviors. The security agent is designed for the purpose of monitoring bidding agents for their activities, and detecting shilling behaviors based on shill patterns and security policies. Since it is not feasible to monitor every activity of each agent in details, we decrease the load of the security agent by defining different security levels such that the depth of checking is directly proportional to the level of distrust in the user. For example, a bidding agent with security level of 1 will receive the most careful monitoring.

The following *SA Policy* is an example of detection rules that defines the way of monitoring a bidding agent with security level 2.

```
% Invoked if the security status is level 2
security_level_2(SHILL_STATUS,SHILL_PROB,DIFF_IN_
LOC,CONC_BID,WL_RATIO,PRESENT_INITIAL_STYLE):-
proximity_of_ip(TEMP1,DIFF_IN_LOC),
concurrent_bid_check(TEMP2,CONC_BID,WL_RATIO),
initial_bid_style(TEMP3,PRESENT_INITIAL_STYLE),
SHILL_PROB is TEMP1+TEMP2+TEMP3,
status_evaluation(SHILL_STATUS,SHILL_PROB).
```

To detect shilling, the security agent is configured to perform different types of security checks. At the lowest level (level 4), only the distance in locations of a buyer and a seller are checked according to their IP addresses. At level 3, we check if a buyer is participating in concurrent auctions with identical auctioned items. Note that concurrent shilling, where a bidding agent places bids on an auction item with higher auction price rather than on the auctioned item with lower auction price, is a strong indication of shilling behaviors. At level 2, the security agent analyses the bidding style of a buyer against common shill patterns. In many cases, it has been found that a shilling agent does aggressive biddings at the beginning, and stops bidding towards the end of the auction to avoid winning the auction. Finally, at the highest security level (level 1), the security agent performs all above checks coupled with an analysis of the bidding agent's history. The security agent derives a shill factor by applying different security rules on the agent's current and previous behaviors. If the shill factor is high enough, the agent's bidding status will be set as *shilling*, and the state module will be updated. The updated information stored in the state module will be used by the role assignment module when the shill bidder makes a new bidding request. Furthermore, as an alert, the security agent will inform all participating agents about the detected abnormal behavior. In a severe situation, when an agent's shilling behaviors are committed based on strong evidence, the security agent will force the auction to be closed and notify all involved users about such decision.

5. An Example

Our approach can be illustrated by an example of online auctions that involve shilling behaviors. In our example, we consider two concurrent online auctions – we call them *Auction 1* and *Auction 2*, which are initiated by seller *S1* and *S2*, respectively. The auctioned items are “Nikon 8x Optical and 4x Digital Zoom Camera,” which are identical in both of the two auctions. There are five bidding agents *B1* to *B5* that may put bids on either of the auctions. Agent *B1*, *B2* and *B4* are configured to work as normal bidders. But for agent *B3*, we set up a bad feedback history for the agent initially. Consequently, the role assignment module downgrades *B3*'s role from *average* buyer to *untrusted* buyer when *B3* makes a

bidding request, and the access control module sets agent *B3*'s security status to level 1. Furthermore, we configure agent *B3* with the following bidding strategy: it tries to drive up the auction price of the item listed by seller *S2* aggressively at the beginning, but stops to put bids on that item when the auction price reaches a certain value. In addition, agent *B5* is configured with a strategy called *Preferred Seller Strategy*, which instructs agent *B5* to put bids on the item listed by seller *S1* rather than *S2* for most of the time. Table 1 lists the role assignment and some of the access rights for each bidding agent.

Table 1. State information of bidding agents

Bidding Agent	Previous Role	Role Assignment	Access Control
B1	most trusted buyer	most trusted buyer	no actions Sec_status: level 4
B2	trusted buyer	trusted buyer	no actions sec_status: level 3
B3	average buyer	untrusted buyer	warning sec_status: level 1
B4	average buyer	average buyer	no actions sec_status: level 2
B5	average buyer	average buyer	no actions sec_status: level 2

While both *Auction 1* and *Auction 2* are running, the security agent monitors each bidding agent according to its security level. Since the bidding agent *B1*, *B2* and *B4* show their normal bidding behaviors, the security agent sets their bidding status as *normal*. On the other hand, since agent *B5* puts bids on both of the auctions, and the security agent detects that *B5* sometimes bids on one of auctions with higher auction price. By further analyzing *B5*'s bidding behaviors, the security agent has found that *B5* bids on the item listed by seller *S1* for most of the time, and puts bids on the item after the reserve price has reached. This indicates that agent *B5* does not attempt to drive up the price because it has no intention to avoid winning the auction. Thus the security agent concludes that agent *B5*'s bidding status is *normal*.

Since *B3*'s security status has been set to level 1, the security agent analyzes *B3*'s bidding activities thoroughly and finds that *B3*'s bidding behavior matches the concurrent shilling pattern, where an agent places bids on the auctioned item with higher auction price rather than on the auctioned item with lower price, and also tries to avoid winning an auction by stopping bidding when the price reaches the reserve price. Furthermore, the security agent analyzes *B3*'s current and past bidding transactions as well as the number of wins in auctions listed by both seller *S1* and *S2*. The security agent notices that *B3*'s win-loss ratios on auctions listed by seller *S1* and *S2* are close to 0. Based on the above knowledge, the security agent assigns *B3*'s bidding status as *shilling*. The security agent then notifies all participating agents, and updates the state

module information for agent *B3*. Figure 4 shows the user interface for agent *B3* with a notification from the security agent.

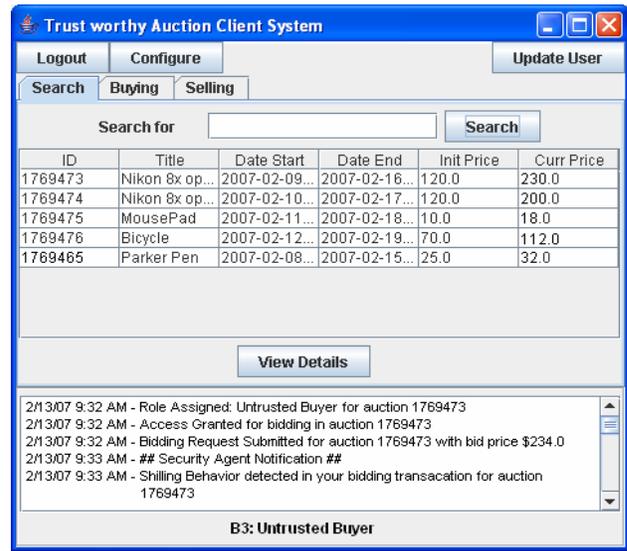


Figure 4 User interface for bidding agent *B3*

When agent *B3* places a new bidding request, the role assignment module assigns *B3* a role of *most untrusted* buyer, and as a penalty, the access control module restricts *B3* from putting bids for a week. Table 2 shows the updated state information of each agent after the analysis is done by the security agent.

Table 2. Updated state information of bidding agents

Bidding Agent	Bidding Status	Role Assignment	Access Control
B1	normal	most trusted buyer	no actions sec_status: level 4
B2	normal	trusted buyer	no actions sec_status: level 3
B3	shilling	most untrusted buyer	one week penalty sec_status: level 1
B4	normal	average buyer	no actions sec_status: level 2
B5	normal	average buyer	no actions sec_status: level 2

In Figure 5, we show a user interface for an auction house administrator to view all auction related activities performed by bidding agents, as well as any actions taken by the security agent.

In our prototype agent-based online auction system, actions against a shill bidder are taken in real-time by updating the agent's role assignment and restricting the agent's access to auction related activities. To ensure a more accurate detection of shill bidders, the security agent also requires evidence such as user's IP address, ratings, user feedbacks and current and past trading histories. In addition, expert experience and considerations of

practical situations are vital for us to set up effective policy rules for skill detection. With more and more expert knowledge on skill patterns [15, 16], our approach can be very effective in skill detection for practical agent-based online auction systems.

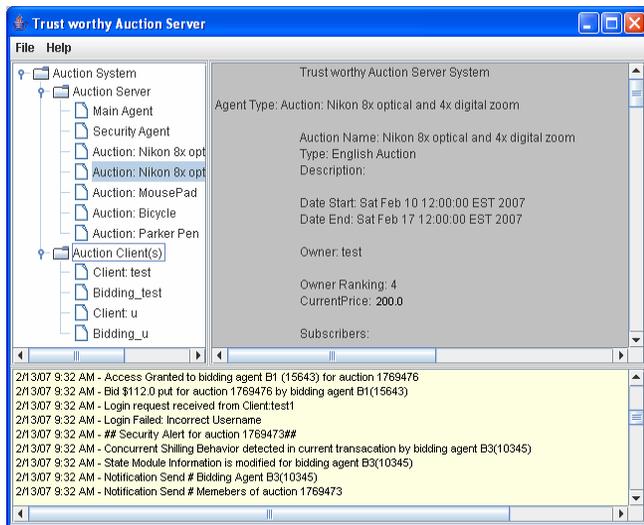


Figure 5 User interface for auction house administrator

6. Conclusions and Future Work

In order to build a trustworthy agent-based online auction system, we introduced a real-time trust management module (TMM) to restrict and prevent undesired bidding behaviors such as shilling behaviors in online auctions. Based on an agent's current and previous behaviors in agent-based online auctions, the real-time trust management module can assign agent roles dynamically, and grant or deny an agent for varying levels of access to auction related resources and activities. Meanwhile, any undesirable bidding behaviors performed by a bidding agent can be automatically detected by a security agent. We have defined different policy rules in Prolog for dynamic role assignment, access control mechanisms, and undesirable bidding behavior detection. The skill detection example, which is simulated on our prototype agent based online auction system, shows that our approach is feasible and efficient. For our future work, we will try to formalize various policy rules, and based on existing work on skill patterns [2, 16], we will try to develop a more accurate method for real-time skill detection in agent-based online auction systems.

References

[1] A. Vakali, L. Angelis, D. Pournara, "Internet Based Auctions: A Survey on Models and Applications," *ACM SIG on E-commerce Exchanges*, Vol. 2, No. 2, Jun 2001, pp. 5-13.

[2] H. Xu and Y. Cheng, "Model Checking Bidding Behaviors in Internet Concurrent Auctions," To appear in *International Journal of Computer Systems Science & Engineering (IJCSSE)*, 2007.

[3] Katia Sycara, "MultiAgent Systems," *AI Magazine*, Vol. 19, No. 2, Summer 1998, pp. 79-92.

[4] TimesOnline, "Revealed: How eBay Sellers Fix Auctions," *The Sunday Times*, Tech & Web, Jan 28, 2007. Retrieved on January 29, 2007, from <http://technology.timesonline.co.uk/tol/news/>

[5] T. Ito, N. Fukuta, T. Shintani, K. Sycara, "BiddingBot: A Multiagent Support System for Cooperative Bidding in Multiple Auctions," In *Proceedings of the Fourth International Conference on MultiAgent Systems*, July, 2000, pp. 399-400.

[6] E. Ogston and S. Vassiliadis, "A Peer-to-Peer Agent Auction," In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002)*, pp. 151-159.

[7] J. Collins, W. Ketter, M. Gini, "A Multi-Agent Negotiation Testbed for Contracting Tasks with Temporal and Precedence Constraints," *International Journal of Electronic Commerce*, 7(1):35-57, 2002.

[8] Bhavani Thuraisinham, "Trust Management in a Distributed Environment," In *Proceedings of the 29th COMPSAX'05*, 2005.

[9] T. Gradison, M. Sloman, "A Survey of Trust in Internet Applications," *IEEE Communications Surveys*, Fourth quarter 2000.

[10] G. Zacharia and P. Maes, "Trust Management through Reputation Mechanisms," *Applied Artificial Intelligence*, 14:881-908, 2000.

[11] A. Herzberg, Y. Mass, J. Mihaeli, D. Naor, and Y. Ravid, "Access Control Meets Public Key Infrastructure," *IEEE Symposium on Security and Privacy*, Oakland, California, USA, 2000, pp. 2-14.

[12] H. Mouri, Y. Takata and H. Seki, "A Formal Model for Stateful Trust Management Systems," In *Proceedings of IASTED International Conference on Software Engineering and Applications (SEA 2005)*, Phoenix, USA, Nov. 2005, pp. 87-92.

[13] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-Based Access Control Models," *IEEE Computer*, 29(2):38-47, 1996.

[14] P. Moraitis and N. Spanoudakis. "Combining Gaia and JADE for Multi-Agent Systems Development," In *Proceedings of the 17th European Meeting on Cybernetics and Systems Research (EMCSR 2004)*, Vienna, Austria, April 2004.

[15] D. H. Chau, S. Pandit and C. Faloutsos, "Detecting Fraudulent Personalities in Networks of Online Auctioneers," *PKDD 2006*, Berlin Germany.

[16] J. Trevathan and W. Read, "Undesirable and Fraudulent Behaviour in Online Auctions," In *Proceedings of the International Conference on Security and Cryptography*, 2006, pp. 450-458.