MAGMA: An Agent-Based Virtual Market for Electronic Commerce

Maksim Tsvetovatyy
tsvetova@cs.umn.edu

Maria Gini
gini@cs.umn.edu

Bamshad Mobasher
mobasher@cs.umn.edu

Zbigniew Wieckowski
wieckows@cs.umn.edu

Department of Computer Science
University of Minnesota, Minneapolis, MN 55455

Abstract

In this paper, we propose an architecture for an agent-based virtual market that includes all elements required for simulating a real market. These elements include a communication infrastructure, mechanisms for storage and transfer of goods, banking and monetary transactions, and economic mechanisms for direct or brokered producer-consumer transactions. We report findings that resulted from implementing and conducting experiments with a free-market agent architecture (MAGMA). MAGMA is an extensible architecture that provides all services essential to agent-based commercial activities. These services are available through an open-standard messaging API, which allows using a heterogeneous set of agents, independently of the platform and language.

1 Introduction

Development of the Internet has spurred a number of attempts to create a virtual marketplace, where agents, both human-controlled and equipped with intelligent algorithms, can participate in trading of physical and electronic goods, as well as stocks and other investment vehicles [Borenstein, 1996, Mullen and Wellman, 1996]. Many researchers as well as commercial companies have created intelligent agent-based systems that support various aspects of electronic commerce such as online shopping and virtual catalogs.

For example, currently there are several intelligent agent-based online shopping services [Chavez and Maes, 1996, Geddis *et al.*, 1995, Doorenbos *et al.*, 1996, Shardanand and Maes, 1995]. The most well established are Bargain Finder, developed by Andersen Consulting as part of their Smart Store Virtual initiative, and FireFly, from Agents, Inc. Both systems assist users in shopping for music CDs and tapes.

Bargain Finder (see `http://bf.cstar.ac.com`) is an agent that searches several online music stores for lowest prices on CDs and cassettes. The system is essentially a database search engine. Shopping is limited to retailers that subscribe and pay for the customer referrals. Also, customers have to know and spell correctly the names of both the artist and the album - so the system is more oriented toward a user that knows what he or she wants than to a casual shopper.

FireFly (see `http://www.agents-inc.com/`) is an agent-based mail order outlet. Through surveys and ratings, the system attempts to establish the user's preferences in music styles and artists and offer the user a list of CDs that conform to these preferences. This system worked very well after a few minutes of training (entering and rating CDs), returning a complete set of Miles Davis albums after we entered a reference to John Coltrane, both jazz artists. FireFly is a fine collection browser, but it is limited to one retailer and does not have any way of comparison-shopping.

While both of these systems provide interesting shopping experiences, they and several other similar systems, fall short of a market metaphor, since they do not include all of the infrastructure and mechanisms necessary for electronic commerce. In addition, these systems generally lack facilities for automated purchasing and agent cooperation.

Kasbah [Chavez and Maes, 1996] is a Web-based system where users can create autonomous agents that buy and sell goods on their behalf. The system uses a classified ad metaphor, where agents post their offers to the common blackboard and then lie dormant until there is a response. While this system has a practical advantage of not requiring users to stay logged on all the time, the system has several shortcomings that prevent it from being a virtual market.

Kasbah agents use a very limited message protocol that does not allow for interagent negotiations, as well as specialized agents such as banks or advertisers. It is a proprietary server-side system, which makes it impossible to use heterogeneous mix of agents in the market. By making all agents reside inside one Lisp image on the server, the authors of Kasbah bypassed the issue of providing secure TCP/IP communications. However, this also limits the capabilities of the system. While Kasbah is a very plausible solution for a privately-run commercial web server, it is not a complete or scalable virtual market.

The Michigan AuctionBot (see `http://auction.eecs.umich.edu/`) is an experimental Internet auction server, whose main purpose is to allow people to run an auction over the net. The AuctionBot is only an information service. It collects bids, determines the results of the auction using a well-defined set of auction rules, and notifies the participants. It does not execute transactions or enforce exchanges.

The architecture we propose for a marketplace includes the infrastructure required for conducting commerce on the Internet, supports communication among agents, and allows for various forms of automated and human-controlled transactions.

In our research, we focused on the following issues:

- What is a good architecture for an agent-based marketplace;
- What infrastructure elements are needed to serve an agent-based marketplace, and how can they be implemented;
- What are appropriate economic mechanisms and transaction protocols for an agent-based virtual market.

To answer these questions, we have implemented a prototype of an agent marketplace architecture, named MAGMA (Minnesota AGent Marketplace Architecture), and completed a series of experiments with it, determining its strengths and weaknesses, as well as directions for future development.

The paper is organized as follows. Section 2 describes the necessary components of an agent-based virtual market system and issues related to designing a general framework for such a system. Section 3 considers appropriate economic mechanisms necessary to support both direct and brokered transactions among consumers and producers engaged in electronic commerce within the virtual market. Section 4 presents the architecture of the MAGMA system, including communication and negotiation protocols and Section 5 discusses the results of our experiments with the prototype through an example. Finally, Section 6 provides a brief overview of future extensions and research directions.

2 Architecture for Agent-Based Marketplace

An agent-based virtual marketplace needs to exhibit many properties attributed to physical marketplaces [Amihud, 1978]. There has to be a banking system to handle the money [Panurach, 1996], some (preferably secure) communication infrastructure [Bhimani, 1996] (but, for a different view see [Borenstein, 1996]), a system that enables agents to transport and store goods securely, means to advertise what is up for sale, economic mechanisms for transactions, as well as administrative and policing systems.

2.1 Banking

In order for an agent-based marketplace to become anything more than a toy, it has to be able to communicate with existing banking and financial services. Thus, the design of a banking system, as well as agent-to-bank communications has to be built on an open standard, allowing traditional banks to integrate a marketplace interface into their legacy systems.

Panurach [Panurach, 1996] suggests several advantages of such an open standard for financial services:

- users could choose among services provided by different financial intermediaries, forcing them to compete in the open market;
- there would be fewer heterogeneous systems that need to be regulated, thus making the implementation of government policies less ambiguous and more effective;
- an open standard providers a more consistent and efficient interface from the users' perspective.

Another crucial element in providing financial services over agent-based systems is security. Consequently, the communication protocols for all communications, especially agent to bank and bank to bank, need to include a layer of encryption, as well as other safeguards [Bhimani, 1996].

2.2 Communication Infrastructure

It is important to have an efficient and robust communication infrastructure built into the core of the system. For example, such system should not rely on a central hub to route messages, but instead have a mesh of redundant hubs, interconnected with each other. This system will be somewhat similar to the Internet email system, and, in fact, can even use the existing facilities.

A communication system has to be built on an open standard to enable developers to produce platform-independent systems that plug into the marketplace architecture. There are some design and communication protocol constraints that must be followed:

- Agents must be able to access global bulletin boards that contain offers to buy and sell from other agents;

- Agents must use a common language for all outbound communications. Minimal language must include posting and responding to offers, negotiating and executing transactions. If agents work as a part of the team with managers and other agents, they must be able to communicate with all members of the team.

In addition, a robust communication infrastructure of processing electronic transactions must satisfy some basic security requirements [Bhimani, 1996], including user confidentiality (usually achieved through encryption), data integrity (data sent as part of a transaction should not be modifiable), message authentication (achieved by using digital signatures or certificates), and nonrepudiation (parties should not be able to deny their participation in a transaction after the fact).

2.3 Transfer and Storage of Goods

An important aspect in an electronic market is the representation and handling of physical goods. While these goods can be easily represented by software objects, these objects have to exhibit some of the qualities of their physical counterparts.

For example, these objects have to be copy-protected to ensure that an object cannot be in more than one place at the same time. To prevent theft, objects may be encoded by the owner, making them accessible only to agents authorized by the owner.

Agents should also be able to arrange for physical shipment of items purchased or provide a "raincheck" that will enable the buyer to have the item shipped at a later time. Electronic items, such as software or results of database queries, are easier to handle and could be sent directly to the buyer using an existing protocol, such as FTP or HTTP.

The existence of rainchecks can create an interesting trading paradigm similar to that of futures trading in commodity markets. An agent can profit by buying a raincheck when the price is low and selling it to another agent or shipping the item later, when the price has risen.

2.4 Advertising

Advertising is one of the keystone elements of a free market. Thus, a virtual free-market system has to adopt some form of it. However, in a market populated by agents, the visual appeal, slogans or other methods that use human psychology will not work, prompting the developers to find alternative ways to advertise to an audience of software agents.

One method of creating an advertising system is to create one or more advertising agents that maintain local databases of items for sale and respond to query messages sent by the buyer or seller agents. These agents could also perform filtering or custom queries.

Another approach was described by A. Keller [Keller, 1995]. In this approach, a *facilitator agent*, armed with a knowledge base, acts as a broker, querying other advertising agents and databases for information requested by its client (which could be another software agent). The facilitator agents are also responsible for translating and routing information to other agents. Facilitators can charge for their services and provide a rich, but uniform content environment that would make choosing a bid to follow up a much more informed decision.

3 Economic Mechanisms for Electronic Commerce

Our goal is to design a general framework which deals, in an efficient and uniform manner, with various activities involved in electronic commerce, including buying and selling, as well as banking, monetary transactions and advertising. In the previous section we discussed this general framework in terms of its components. In this section, we analyze, in more detail, the economic transaction models of electronic commerce and how they could be used effectively in the design of agent-based market systems.

3.1 Transaction Models for A Virtual Market

Recently, there has been a substantial amount of effort devoted to the study and design of transaction models for electronic commerce [Panurach, 1996]. Most of the proposed models have fallen under the following two general categories:

1. direct transactions among providers and consumers;

2. agent-based brokered transactions.

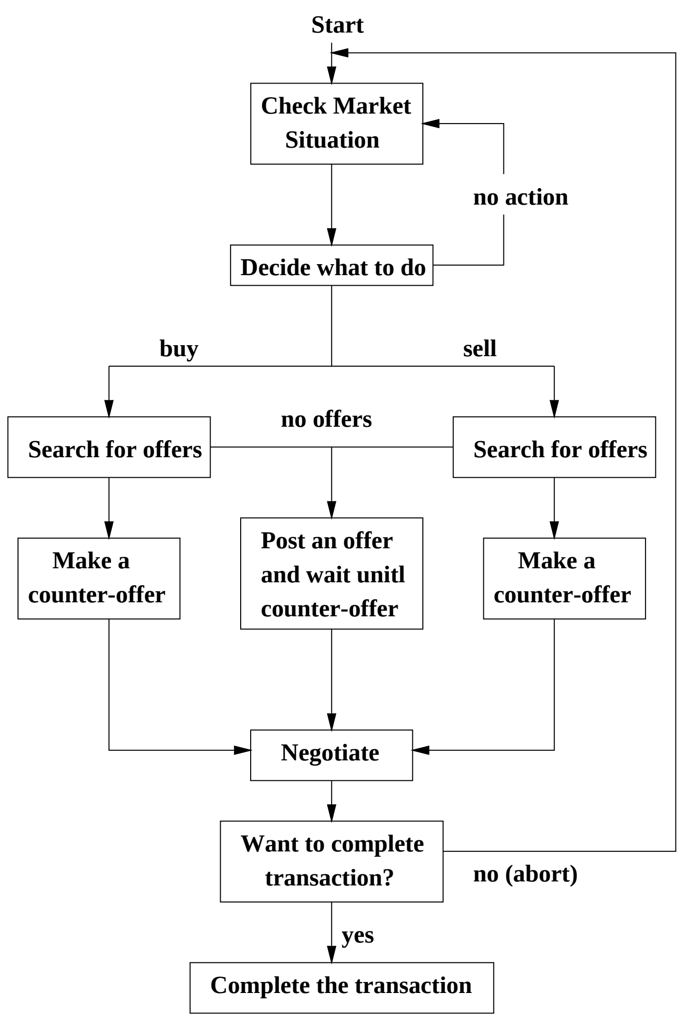

Figure 1: Algorithm for direct transactions between agents.

The direct transaction model has the advantage that purchases can be privately negotiated. Producers can send advertisements to customers either directly or through online catalogs, and customers can select from among the available products. Providers and consumers can suggest prices and some negotiation and bargaining may be possible.

Intelligent software agents may or may not be involved in the direct transaction model. As noted earlier, several commercial and non-commercial online shopping services have been developed recently which involve the use of intelligent agents to assist consumers in finding bargains, facilitate negotiation among providers and consumers, or provide help in locating appropriate items over a distributed electronic market (e.g., the Internet). Some examples of these include the Bargain Finder [Krulwich, 1995], FireFly [Shardanand and Maes, 1995], Infomaster [Geddis *et al.*, 1995], Kasbah [Chavez and Maes, 1996], and ShopBot [Doorenbos *et al.*, 1996].

A generalized virtual market framework, however, needs to incorporate mechanisms to facilitate the investigation of market conditions and to assist users through a variety of decision algorithms. A general agent algorithm for the direct transaction model, within the framework of our agent-based virtual market, is depicted in Figure 1.

Any realistic virtual market framework, such as the one we propose in the MAGMA system, must include mechanisms for direct transactions and negotiation. However, there are many situations where such direct negotiation of exchanges is not desirable. For example, it may be expensive for providers and consumers to find each other in a distributed market system; or either may wish to remain anonymous; or users may wish to relegate the time consuming task of negotiating for the best price to intelligent agents that work on their behalf.

In such situations the brokered transaction model may be more suitable for electronic commerce. In a virtual market framework, such brokered transaction must be facilitated by automated agents that engage in bargaining and/or bidding to find the "best" deals (the highest bid, from the perspective of the provider; or the lowest bid, from the perspective of the consumer).

In recent work on automated bargaining agents, researchers have proposed a variety of bargaining strategies and economic mechanisms suitable for brokered transactions. A realistic virtual market framework must also provide economic mechanisms necessary for this type of automated brokered transactions. Figure 2 depicts a general agent algorithm for the brokered transaction model, within the framework of our agent-based virtual market.

3.2 Rules and Strategies for Agent Brokered Transactions

Design of effective economic mechanisms for agent transactions depends heavily on the model used to describe the interaction of participants. Researchers in distributed AI and in Economics have generally used game theoretic models to describe the behavior of transaction participants. From this point of view, the players (i.e., the buying and selling agents) provide bids on a particular item and engage in negotiations according to their respective bargaining strategies to arrive at a "fair" price for the item. Each player's *strategy* guides its actions at various steps in the game based on the available information.

There are many considerations that come into play with automated bargaining agents engaged in bidding processes. If an unrestricted bidding protocol, such as the ones suggested in [Chaterjee and Samuelson, 1988], is employed, or if the transaction model used is an agent-based direct transaction model, then it may be difficult for any particular agent to determine the "fair" price for the transaction.

The problem is that, generally, the agents will not have enough information about the true valuation of the item ascribed by other (possibly dishonest) agents. In such a protocol, agents may have to engage in a variety of strategic behaviors to either reveal the private information held by competing agents or to guard against dynamic strategies that can extract their own private information.

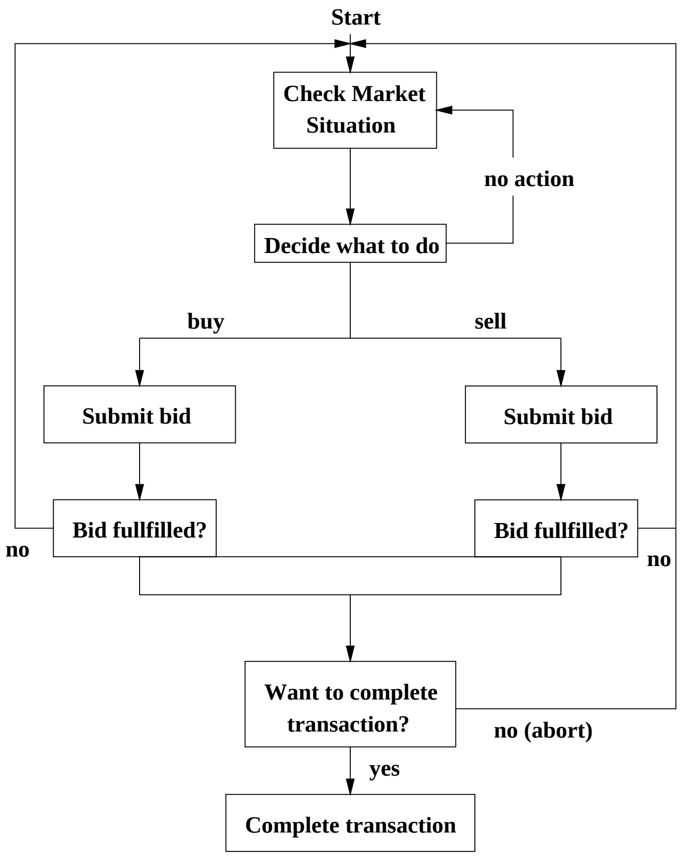

Figure 2: General form of the algorithm for brokered transactions among agents. Agents submit their bids to a broker agent (not shown in the figure). Different bargaining strategies can be used for negotiation.

Each agent, i, knows at least, the *reservation price* of an item I for its owner. The reservation price is the maximum willingness-to-pay (for a buying agent), denoted by $v_i^b(I)$, or the minimum willingness-to-sell (for a selling agent), denoted by $v_i^s(I)$. Now, for example, if a selling agent i can learn the reservation price for a buying agent j, it can make the buyer a take-it-or-leave-it offer at (possibly slightly higher than) $v_i^b(I)$. On the other hand, if a buying agent j learns $v_j^s(I)$, a similar situation may occur.

It may also be beneficial for agents to lie about their reservation price, in order to receive the maximum pay-off. For example, agent j may bid much higher than $v_j^s(I)$, in which case the proper course of action for the agent i would be to give a counter offer between the current bid and its $v_i^b(I)$. In a market situation, this can get substantially more complicated since, there will be other buyers and sellers willing to exchange the same item. For instance, if agent j bids much higher than $v_j^s(I)$, it may loose the bid to another selling agent j which bids in the gap between $v_j^s(I)$ and $v_i^b(I)$. Clearly, the buying agent may also face a similar situation in trying to decide on an appropriate counter offer. A good example of this kind of situation is the telephone companies selling their long distance services to potential customers [Rosenschein and Zlotkin, 1994]. A great deal of resources can be expended in back and forth negotiation or in strategies to obtain private information from buyers (e.g, their reservation price) by analyzing their bidding patterns, etc.

As noted earlier, [Sun and Weld, 1995] considers several bargaining strategies for automated agents and concludes, empirically, that more complex, adaptive strategies perform better than simple discounting strategies for bargaining. Other bargaining protocols have also been suggested which involve different degrees of overhead in terms of strategic behavior. For example, one possibility is to use a method similar to the standard *English auction*, where a seller (the auctioneer) continuously raises the price until only one bidder remains. Alternatively, the buyer can start from a high price and continuously lower it until a seller is found. This kind of mechanism could be highly iterative with a great deal of communication overhead in a distributed environment.

Another method that could be used is based on the *sealed bid auction* model. In this model, each buying agent submits a sealed bid known only to the seller (or the auctioneer). The buyer with the highest bid is awarded the item at the bid price. The situation is reversed for selling agents. The advantage of this model is that it is a one-step procedure, however it requires strategic reasoning, about the "beliefs" of other participants and how they will bid.

These and other similar strategies have certain common characteristics which may make them unattractive in the design of economic mechanisms for automated agents:

- they require a great deal of communication, which could be expensive;
- they, generally, involve iterative processes to allow for negotiated bidding;
- they require a high degree of security in order to keep the agents private information secret; and
- they involve highly dynamic strategies (possibly including reasoning about the beliefs of other agents) for extracting the true evaluation of items by other buyers and sellers.

An important question is whether there is an economic mechanism (a bargaining protocol) for agent-based brokered transactions that is simpler, more efficient, and more stable, and that can achieve everything the aforementioned "unrestricted" protocols can achieve. It turns out that, in domains where the buying and selling of goods and services can be modelled by *private-value auctions* (i.e., where an agent's reservation price can be totally determined locally and is independent from other agents' reservation prices), the answer to this question is "yes." The *Vickrey mechanism* [Vickrey, 1961] is one of such protocol which promotes truthful bidding among self-interested agents and thus avoids the need for counterspeculation. Below, we provide a brief description of the Vickrey mechanism. For more detailed discussion we refer the reader to [Rosenschein and Zlotkin, 1994, Varian, 1995].

3.3 The Vickrey Mechanism

The Vickrey mechanism is similar to the sealed-bid mechanism described earlier; the difference is that the settlement price for the transaction will be the second lowest bid (from among the bids by selling agents) or the second highest bid (from among the bids by buying agents).

For example, suppose that a selling agent i wants to sell an item I. Buyers j and k bid at their reservation price (maximum willingness-to-pay), $v_j^b(I)$ and $v_k^b(I)$, respectively. The item is then awarded to the buyer with the highest bid, but at the price paid by that buyer is that of the second highest bid. So, if $v_j^b(I) \geq v_k^b(I)$, then the agent j is awarded the item with the price $v_k^b(I)$. In the case of a buying agent trying to buy an item (e.g, long distance service), the selling agents will bid at their reservation price (minimum willingness-to-sell), and the sale is awarded to the seller with the lowest bid, but at the price of the second lowest bid.

Note that under this mechanism, the agents are motivated to tell the truth about their reservation prices. For example, in the case of selling agents, a seller i will not bid lower than $v_i^s(I)$, since other sellers may bid higher and thus forcing i to sell at lower than its acceptable minimum price. The agent i will not bid higher than $v_i^s(I)$, since that will increase the probability that it will not be awarded the sale (other agents may bid lower). It is also noteworthy that the buyers in this scheme pay a premium (the difference between the lowest and the second lowest bid) to guarantee that rational selling agents tell the truth about their reservation price, but in a market situation with many sellers, competition forces this margin to be reduced proportionally. Clearly, in the case of buying agents a dual situation will exist.

There are many advantages to using the above mechanism in the design of automated brokered transaction models for electronic commerce:

- provides a simple and stable mechanism for automated transactions;
- requires no iterative negotiation strategies;
- requires minimal security features, since agents need not hide their reservation prices;
- does not require dynamic strategic behavior to reveal other agents private information, since truth telling is the dominant strategy.

Furthermore, it has been shown that this mechanism can achieve the same utility for the agents involved as other less direct mechanisms where information communicated is something other than the true reservation price [Huberman and Hogg, 1995, Rosenschein and Zlotkin, 1994, Varian, 1995]. In other words, there is no loss of generality in restricting the transaction protocols to this type of "direct revelation" mechanism.

The simple Vickrey mechanism described above is a special case of the *Generalized Vickrey Mechanism* or GVM. The GVM is capable of handling many more complex problems than buying and selling, including a variety of resource problems, problems involving goods with multiple units, and problems involving "collective" or "public" goods. For further discussion of this topic see [Varian, 1995].

While very simple yet powerful mechanism, the Vickrey mechanism may not be appropriate in all domains. For example, Sandholm and Lesser [Sandholm and Lesser, 1995] have also argued that Vickrey mechanism is inadequate for domains where agents are *bounded rational*, i.e., where agents' valuations are affected by time or by costly computational resources. Also in domains where an agent's marginal costs (and thus its reservation price) are determined by other agents' valuations, truthful bidding will not necessarily be the dominant strategy. This is the situation in *public-value auctions* such as the stock market.

Furthermore, even with the Vickrey mechanism as the underlying protocol, it may be necessary for agents to consider the impact of their valuation and the settlement price on the marginal cost and the reservation price of related tasks that could be negotiated in the future. This type of activity (future valuation based on present reservation prices) can be subject to strategic behavior and counterspeculation.

Vickrey mechanism also assumes that the (selling) agents' utility is maximized by selling at higher than its reservation price. In some domains this can be too restrictive of an assumption. For example, an agent could resort to strategic behavior if its goal is to increase market share by selling low. A similar situation occurs when agents can commit to a transaction at a later time than the bidding phase and are subsequently forced to change their bids over time to encourage a sale or purchase.

Because of these shortcomings, a realistic virtual market metaphor must be flexible enough to incorporate a variety of protocols for automated bargaining and be able to adopt the appropriate protocol for a particular domain.

Our proposed MAGMA system, incorporates mechanism for both direct transaction models (see Figure 1) and the brokered transaction model (Figure 2). Because of its simplicity and the effectiveness, we have currently implemented the brokered transaction model using the simple version of the Vickrey mechanism. However, as the algorithm of Figure 2 suggests, this transaction model in our virtual market framework can incorporate other protocols and economic mechanisms such as the ones suggested in the above discussion.

4 Architecture of the MAGMA system

MAGMA (Minnesota AGent Marketplace Architecture) is a prototype of a virtual marketplace system. The current implementation consists of a relay server written in Allegro Common Lisp and a set of agents written in Java that work over the World Wide Web. The current version of MAGMA is designed to work with electronic goods that can be easily transferred over the Internet.

Currently MAGMA includes multiple *Trader Agents*, an *Advertising Server* and a *Bank*. Trader Agents conduct all their business in the system. They are responsible for buying and selling goods and negotiating prices. The Advertising Server provides a classified advertisement service that includes search and retrieval of ads by category. The Bank provides a set of basic banking services that includes checking accounts, lines of credit and electronic cash. All agents are functionally independent and communicate to each other through socket connections.

To facilitate communication between agents, we created a *Relay Server* that maintains

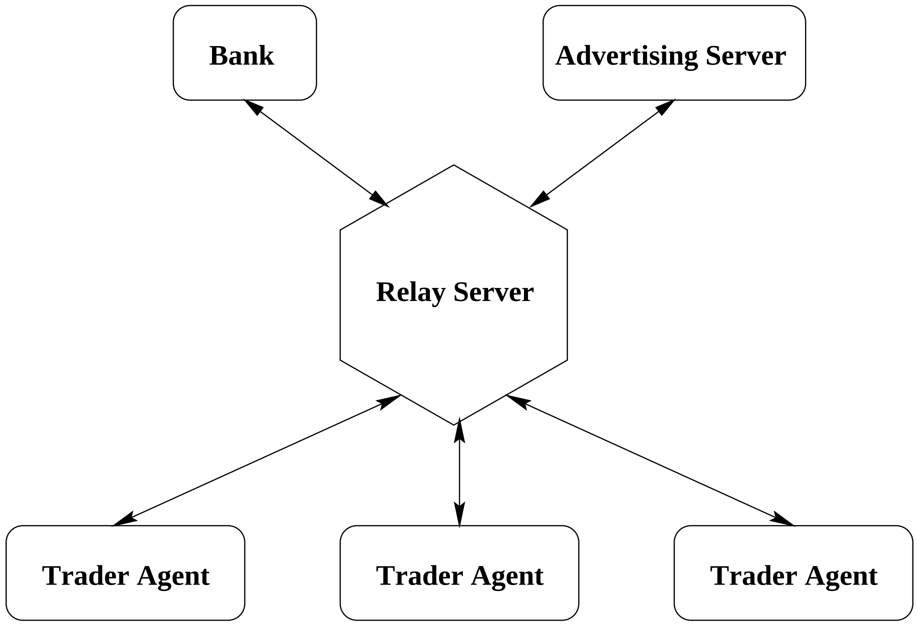

Figure 3: The overall architecture of the agents in the MAGMA system. A Relay Server acts as a central hub for communication among agents. There can be any number of Trader Agents at any time. In our current implementation there is only one Bank and one Advertising Server.

all socket connections and routes messages between agents based on unique agent names. The architecture of the MAGMA system is shown in Figure 3.

While in our current system all agents have similar communication layers and are written in one language, MAGMA is designed to be an open standard, allowing platform and language independent agents that conform to the MAGMA messaging API to connect to the system, register with the Relay Server, and conduct business over the MAGMA infrastructure.

This opens up the possibility of creating MAGMA modules that link up to legacy systems deployed by banks (such as Oracle, Sybase or CICS - which would require an extension to the TCP/IP communication layer), as well as creating "industrial-strength" agent systems for large companies with high volume of sales.

All agents in the current version of MAGMA can be retrofitted with SQL support using the JDBC (Java Data Base Connectivity) protocol. This would enable agents to interface with existing relational inventory databases, as well as banking and virtual catalog databases.

4.1 MAGMA Subsystems

We will now describe in more detail the subsystems of MAGMA, explain their functionality, and discuss the major design decisions.

4.1.1 Communication Layer

All agents share a common communications layer. This insulates them from the physical communication layer, as well as provides an opportunity for transparently adding independent encryption modules.

The communication layer transmits messages as a packet, which consists of the header, that contains names of addressee and sender, the body, that contains message data, and a body checksum at the tail. Packets can be of different length, with parts separated by a reserved character.

Packets are formed from MAGMA objects by using a common object serialization interface. Any object that implements this interface (referred as "Sendable") can be transformed into a packet and sent over the network. Packets are received, verified by using the checksum, and parsed. Then, with the header information stripped off, packets are passed to the de-serialization routine that reconstructs the original object. After the object is restored, it is dispatched to the appropriate handler inside the agent (i.e. money transfers are sent to Wallet, objects related to bids are routed to the Negotiator, etc).

Currently, a number of sendable objects are implemented in MAGMA. They include:

1. Monetary Objects - checks, ecash tokens, payment requests, and messages for communicating with Banks;

2. Widgets - items being bought and sold and requests to send an item;

3. Advertising Objects - advertisements, requests to place an advertisement and status reports from Advertising Server;

4. Bidding Objects - bids, requests for bids, acceptance and rejection notices and all messages used in manual negotiation process.

Trader agents can parse and understand all of the sendable objects, however Banks and Advertising Servers only use a limited set that pertains to their functions.

Due to a security limitation built into the Java language, applets can only make socket connections to the server they were downloaded from. To bypass this limitation, we have created a Relay Server. The Relay Server runs on the same machine as the web server where agents can be downloaded from. It is a multithreaded program designed to maintain multiple socket connections and route messages between them. To achieve that, the Relay Server creates an internal routing table with an entry for every active agent. Packets are routed based on names of agents they are addressed to, in a first-come, first-serve manner. Currently, the Relay Server is written in Common Lisp and can handle up to 20 simultaneous connections. This is sufficient for demonstration and proof-of-concept purposes, however it will require a rewrite in order to make it scalable up to commercial applications.

The Relay Server provides a layer of insulation between agents and the underpinnings of network communication (i.e. IP addresses and ports), as well as added security and potential for mobility and persistence. The current implementation of MAGMA does not include encryption. Encryption will be added in the form of a plug-in module.

4.1.2 Advertising Server

The Advertising Server implements the common "classified advertising" metaphor by interfacing a database of advertisements to a MAGMA messaging API. In the current version of MAGMA, the Advertising Server uses a simple database indexed by category and a compact but efficient search engine. The database contains advertisements that include search keywords, a short description of the product, price, and a URL link to a page with additional information for the buyer.

Currently MAGMA only supports a single Advertising Server, but could be extended to allow multiple Advertising Servers to compete for business from Trader Agents.

4.1.3 Bank

The banking system implemented in MAGMA is similar to the electronic cash system developed by DigiCash, Inc. (see `http://digicash.com/publish/ecash_intro/ecash_intro.html`). When an agent needs to make a payment, it sends a request to its bank to withdraw funds, and receives a secure wrapper, called a check. Before sending out a check, the bank verifies the existence of sufficient funds in the agent's account. Checks can either be written out to a particular agent or be anonymous, providing for a cash-like exchange of money. Another cash-like quality of a MAGMA check is that it can be split into several anonymous checks of smaller value.

After a check is received from the bank, it is routed to the recipient of the money who can either keep it and use it to pay other agents or send it to its bank to be deposited. After receiving such check, the bank would verify its validity and complete the transaction by transferring money between accounts.

A MAGMA Bank provides this money transfer service, as well as other essential banking services, such as standard checking accounts, and well as lines of credit. Accounts are secured by agent's digital signature, making access to accounts limited to their owner. The Bank also keeps audit trails that contain information on all transactions that occurred within a certain period of time (set by the bank operator).

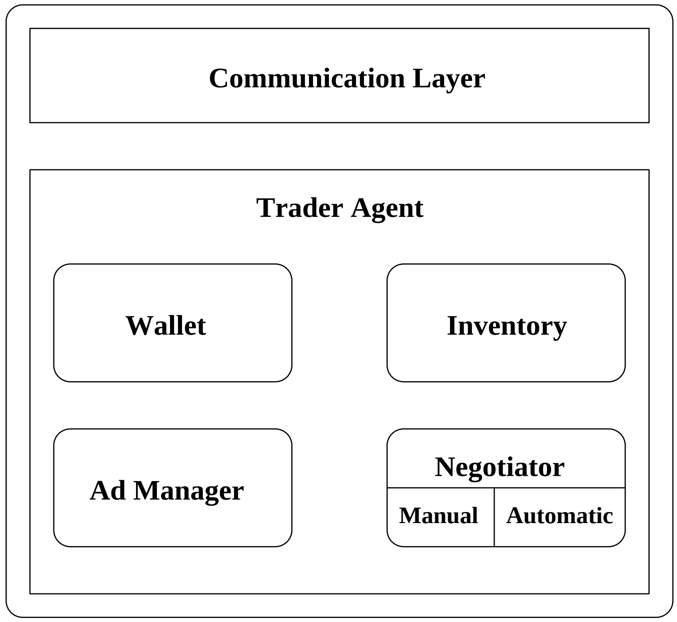

Figure 4: The architecture of a Trader Agent showing its components.

4.1.4 Trader Agents

A MAGMA Trader consists of several interlocking parts: the *Wallet*, the *Inventory*, the *Ad Manager*, and the *Negotiator*. They are shown in Figure 4.

1. The **Wallet** keeps track of all bank accounts, check, tokens, and the operations performed on them. It provides a user interface to the banking services, and automates a number of banking-related operations, such as opening and closing accounts, withdrawing, depositing and splitting of checks. The Wallet also facilitates the payment from an agent to another agent during the purchase of goods. An example is shown in Figure 5.

2. The **Inventory** keeps track of all the goods that the Trader Agent owns (or intends to sell). This includes information about the goods and their URL addresses. The information is kept in a searchable database, as well as displayed on the screen. Goods can be added to the inventory either by manually adding a resource to the database,

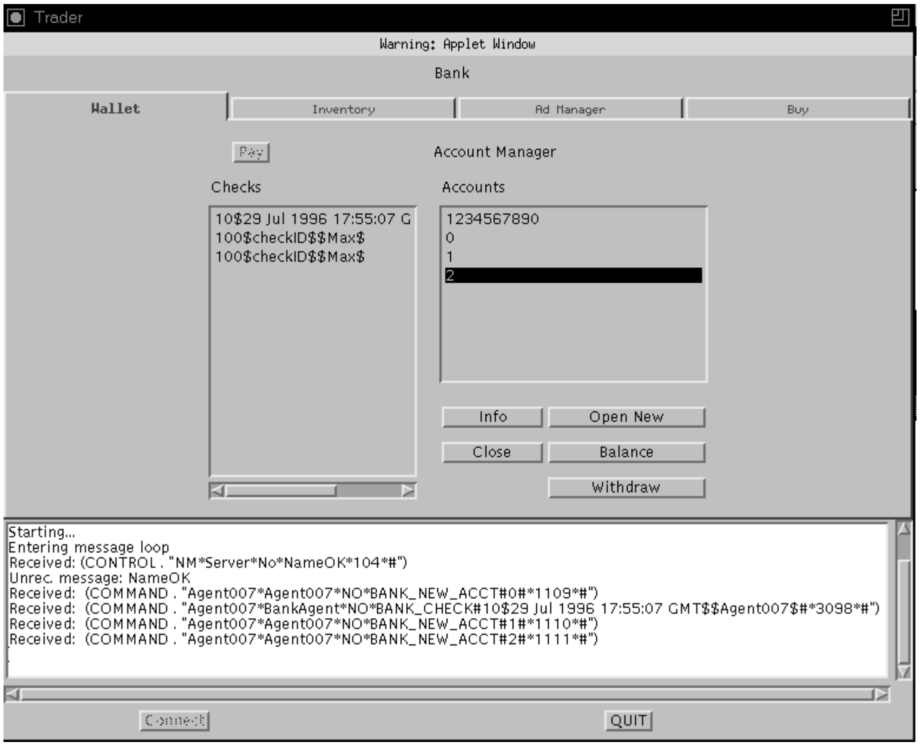

Figure 5: A screenshot of the Wallet tab of a Trader Agent.

or by purchasing goods from other agents. The Inventory passes goods to the Ad Manager and the Negotiator for selling. An example is shown in Figure 6.

3. The **Ad Manager** allows the user to place ads with the Advertising Server to sell goods from the Inventory. It also keeps track of all ads placed and allows the user to remove ads, as well as manually add new ones. An example is shown in Figure 7.

4. The **Negotiator** is responsible for facilitating the search of goods, as well as mediating the negotiations between two manually-driven agents, or for automatically searching for a product to buy and completing the transaction.

   In the manual mode, the Negotiator is mainly used for buying. The Negotiator allows the user to search for goods advertised with the Advertising Server. It passes the query from the user to the Advertising Server, then filters and displays the result. The

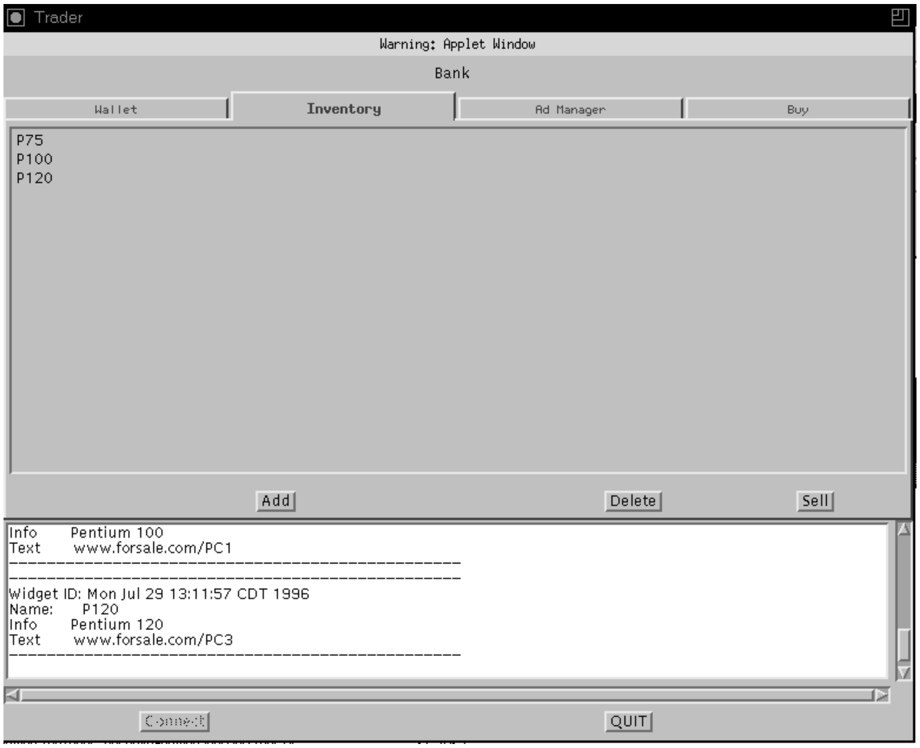

Figure 6: A screenshot of the Inventory tab of a Trader Agent.

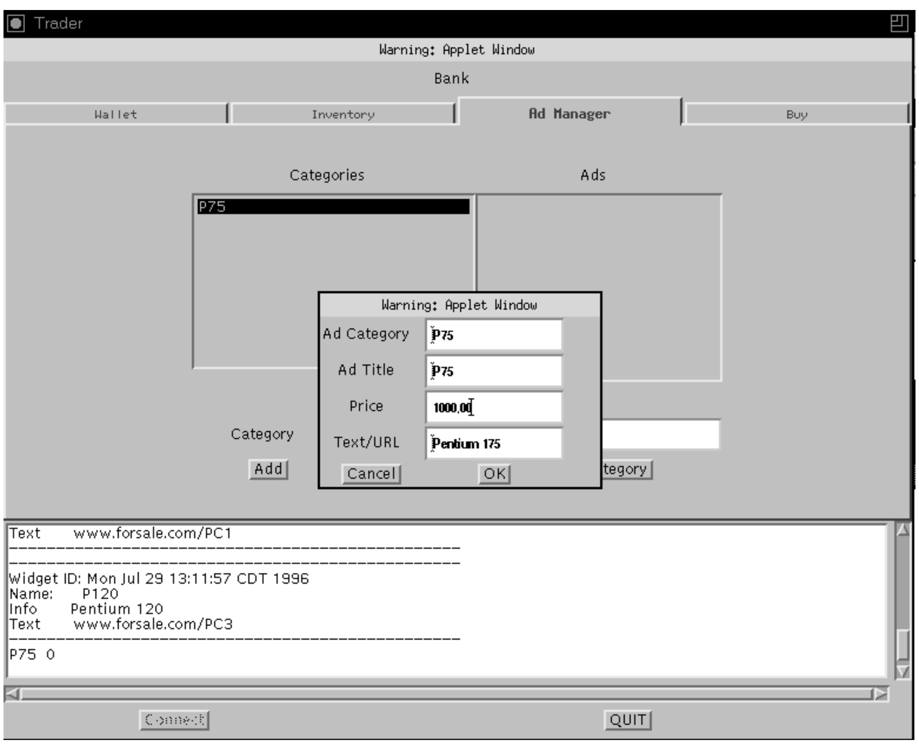

Figure 7: A screenshot of the Ad Manager tab of a Trader Agent.

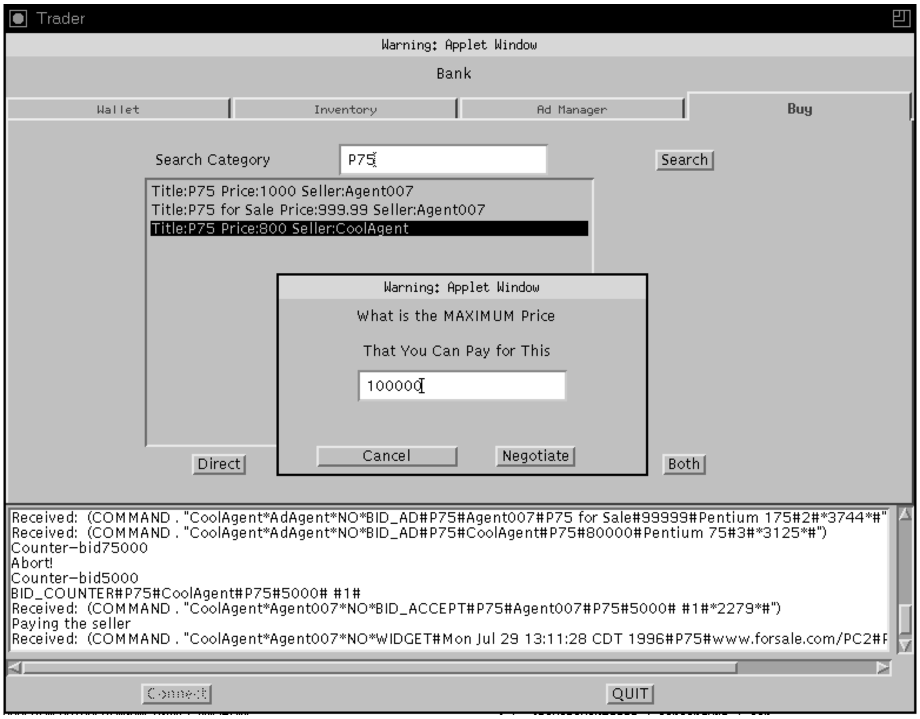

Figure 8: A screenshot taken during a manual negotation between two agents.

user can start manual negotiations with the selling agent, immediately accept one of the offers, or request more information from a Web page and then decide whether to complete the transaction (see algorithm of Figure 1). A screenshot of an example of manual negotiation is shown in Figure 8. Manual negotiation is an interactive process that allows the buyer and the seller to iterate the negotiation until either they reach consensus or decide not to conclude the transaction.

The brokered (automatic) mode can be used for both buying and selling and implements a subset of the Vickrey auction mechanism that was discussed earlier. Trader agents advertise their goods for brokered selling according to rules of the Vickrey auction (i.e. advertising at the lowest price they are willing to accept, or their reservation price). A Trader Agent in buying mode searches for goods listed below the buyer's reservation price. The sale is awarded to the seller with the lowest bid, but at the

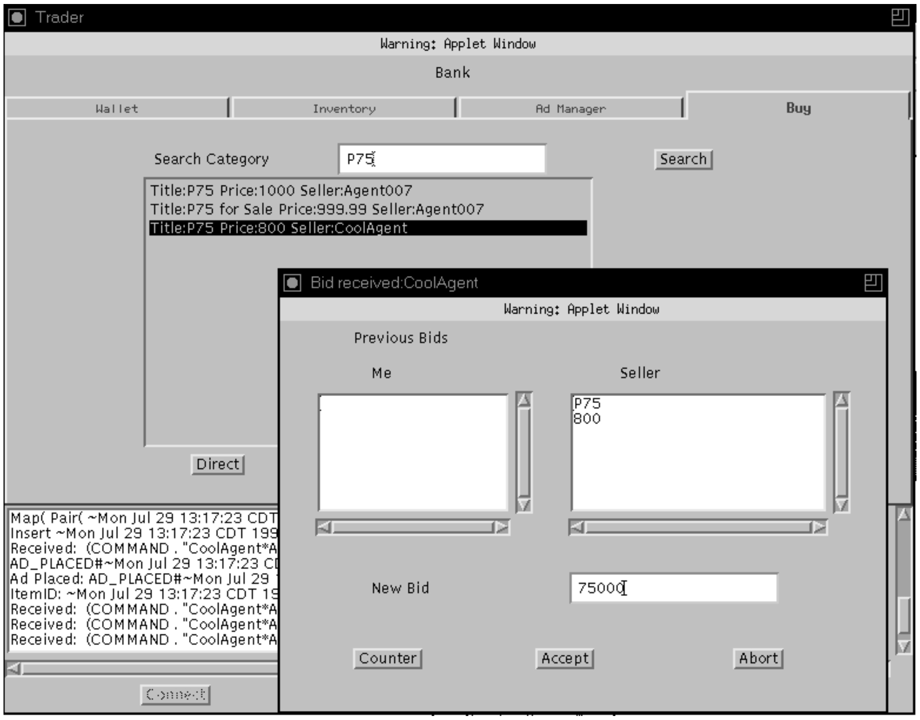

Figure 9: A screenshot taken during an automated transaction between two agents.

second lowest price. The transaction is automatically completed after the buyer is prompted to choose the mode of payment. An example of an automated transaction is shown in Figure 9.

5 Example Transactions

This section describes in detail how a transaction is conducted.

Stage 1 - Initialization

In the first stage, the agents are initialized and prepared to become operational. Even though in this example all agents are started in the first stage, it is possible for agents to join and leave trading at different times, according to their users' desires.

1. The bank accounts are open, and initial deposits are made;
2. The operators of each of the seller agents enter the information about the goods they have for sale into the inventory database;
3. The operators create advertising HTML pages and place them on their Web servers;

4. Prices for goods are set according to the preferred transaction model (manual negotiation or brokered transaction);

5. Ads containing short descriptions of the goods, prices, and URLs for the Web advertisements are packaged in messages and uploaded to the Advertising Server;

6. The Advertising Server creates an index of the ads received.

Stage 2 - Negotiation

In this stage, agents representing the buyer retrieve the relevant ads from the Advertising Server, choose an offer to follow up on, and negotiate the sale price with the seller agent.

1. **Manual Negotiation**

   (a) The buyer's agent queries the Advertising Server for ads in the category indicated by the buyer;

   (b) The Advertising Server returns a stream of messages containing the ads that match the search criteria given by the buyer;

   (c) The buyer chooses an offer to follow up on and initiates a manual direct negotiation;

   (d) The buyer's counter-offer is routed to the seller agent;

   (e) The human operator of the seller agent decides whether to accept the offer, and replies with a counter-offer or aborts the transaction;

   (f) The operators of the buyer and seller agents exchange counter-offers until an agreement has been reached or the transaction aborted;

   (g) When an offer is accepted, the agents proceed to Stage 3 - completing the transaction.

2. **Automatic negotiation**

   (a) The buyer's agent queries the Advertising Server for ads in the category indicated by the buyer;

   (b) The Advertising Server returns a stream of messages containing the ads that match the search criteria given by the buyer;

   (c) The buyer's agent filters the incoming ads and finds the lowest and second lowest bids. It awards the sale to the lowest bidder, but at the price of the second lowest bid.

Stage 3 - Completing the transaction

1. Upon agreement, the seller's agent sends the message containing information on the location of the purchased goods (URL) and a billing message for the amount of sale;

2. Upon receipt of the billing message the buyer's agent switches to the Wallet panel and offers the user a choice of paying out of one of the bank accounts, or with an electronic cash token;

3. When both item and payment are received, the agents flag the transactions as completed and move them into the history file.

6 Conclusions and Directions for Future Work

In this paper we proposed a general framework for an agent-based virtual market system, which contains all of the essential elements needed to conduct electronic commerce, including mechanisms for banking and financial transactions, a communication infrastructure, facilities for advertising, transfer and storage of goods, and economic mechanisms and transaction protocols. We presented MAGMA, a working prototype based on this general framework.

Currently MAGMA is in an advanced prototype stage. and is under continuous development. This section presents some of the extensions and refinements of the architecture that we are planning for the near future.

One of the major areas of work will be refining and extending the messaging API. One of the possible strategies would be to change the API to conform to a standard Agent Communication Language (ACL). The system might also incorporate the emerging standards for electronic commerce and money transfers (such standards are being developed by Sun in alliance with other major players in the computer and financial industry). Another extension to the communication infrastructure is support for a standard encryption protocol that would enable agents to conduct secure transactions.

We will extend the brokered transaction model to incorporate the generalized Vickrey mechanism that can handle more complex problems, such as a variety of resource problems, bartering, problems involving goods with multiple units (such as wholesale transactions), as well as problems involving "collective" or "public" goods (for example, a pool of software licenses). We will also conduct experiments in the brokered transaction model with protocols other than Vickrey (since the architecture is flexible enough to handle others), for domains were Vickrey is not appropriate (see Section 3.3). Given the open architecture of MAGMA, these extensions can be implemented adding specialized broker agents.

Another possible direction is extending the Advertising Server to support virtual and smart catalogs such as those described by Keller [Keller, 1995], and creating Advertising Agents that would employ market mechanisms to place advertisements in a strategic manner. Advertising Agents could have different fee structures for their services. For example, an Advertising Agent could charge for placing ads, running ads for certain amounts of time or for the quantity of goods for sale by one merchant. Determining the fee structure would be entirely up to the company that runs each Advertising Agent. MAGMA could also be extended to allow multiple banks.

In the near future, we are planning to release the system for public testing on the World Wide Web. Initially the system will support limited forms of consumer-producer transactions, including electronic cash payments for goods. Gradually, it will be expanded to include the full range of mechanisms necessary for electronic commerce in a virtual market. During the initial stages of testing, we will provide free content (such as royalty-free clipart) in exchange for "experimental" MAGMA money handed out to all participants. This will let us avoid potential liabilities involved in real monetary transactions.

References

[Amihud, 1978] Y. Amihud. *Bidding and auctioning for procureement and allocation*. New York University Press, New York, 1978.

[Bhimani, 1996] Anish Bhimani. Securing the commercial Internet. *Comm. of the ACM*, 39(6):29–35, 1996.

[Borenstein, 1996] Nathaniel Borenstein. Perils and pitfalls of practical cybercommerce. *Comm. of the ACM*, 39(6):36–44, 1996.

[Chaterjee and Samuelson, 1988] K. Chaterjee and L. Samuelson. Bargaining under two-sided incomplete information: The unrestricted offers case. *Operations Research*, 36(4):605–618, 1988.

[Chavez and Maes, 1996] Anthony Chavez and Pattie Maes. Kasbah: An agent marketplace for buying and selling goods. In *Proc. of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, UK, April 1996.

[Doorenbos *et al.*, 1996] B. Doorenbos, O. Etzioni, and D. Weld. A scalable comparison-shopping agent for the world-wide web. Technical Report TR96-01-03, University of Washington, 1996.

[Geddis *et al.*, 1995] Donald F. Geddis, Michael R. Genesereth, Arthur M. Keller, and Narinder P. Singh. Infomaster: A virtual information system. Technical report, Intelligent Information Agents Workshop at CIKM '95, December 1995.

[Huberman and Hogg, 1995] B. Huberman and T. Hogg. Distributed computation as an economic system. *Journal of Economic Perspectives*, 9(1):141–152, 1995.

[Keller, 1995] Arthur M. Keller. Smart catalogs and virtual catalogs. In *International Conf. on Frontiers of Electronic Commerce*, October 1995.

[Krulwich, 1995] B. Krulwich. Bargain finder agent prototype. Technical report, Anderson Consulting, 1995.

[Mullen and Wellman, 1996] T. Mullen and M.P. Wellman. Some issues in the design of market-oriented agents. In M. Wooldridge, J. Mueller, and M. Tambe, editors, *Intelligent Agents: Theories, Architectures, and Languages*, volume II. Springer-Verlag, 1996.

[Panurach, 1996] Pativat Panurach. Money in electronic commerce: digital cash, electronic fund transfer, and ecash. *Comm. of the ACM*, 39(6):45–50, 1996.

[Rosenschein and Zlotkin, 1994] Jeffrey Rosenschein and Gilad Zlotkin. Designing conventions for automated negotiation. *AI Magazine*, pages 29–46, Fall 1994.

[Sandholm and Lesser, 1995] T. Sandholm and V. Lesser. On automated contracting in multi-enterprise manufacturing. In *Distributed Enterprise: Advanced Systems and Tools*, pages 33–42, Edinburgh, Scotland, 1995.

[Shardanand and Maes, 1995] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating 'word of mouth'. In *Proc. CHI-95 Conference*, Denver, CO, May 1995. ACM Press.

[Sun and Weld, 1995] Ying Sun and Daniel Weld. Automating bargaining agents (preliminary results). Technical Report UW-CSE-95-01-04, University of Washington, 1995.

[Varian, 1995] H. R. Varian. Economic mechanism design for computerized agents. In *USENIX Workshop on Electronic Commerce*, New York, NY, July 1995.

[Vickrey, 1961] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.