In *Proceedings of the Second USENIX Workshop on Electronic Commerce,*
Oakland, CA, November 1996

Market-Based Negotiation for Digital Library Services

Tracy Mullen and Michael P. Wellman

Artificial Intelligence Laboratory
University of Michigan
1101 Beal Avenue
Ann Arbor, MI 48109-2110 USA
{mullen, wellman}@umich.edu

Abstract

The University of Michigan Digital Library is a large-scale confederation of software agents, providing library content and services to users and each other within a distributed network environment. Allocation of resources and activities to the various agents is determined through a market-based negotiation process, where agents tender offers to buy or sell services, basic resources, and other information goods for specified prices. Generalized auction modules resolve these offers into deals among agents. Viewing each agent as an information consumer or entrepreneur, the digital library as a whole constitutes a virtual economy of information goods and services.

Topics: Electronic libraries, negotiations

Introduction

The University of Michigan Digital Library (UMDL) project (Atkins *et al.* 1996) is a large-scale, multidisciplinary effort to design and build a flexible, scalable infrastructure for rendering library services in a digital networked environment. The aim of UMDL is to provide users with a wealth of information sources and library services, and to provide developers with a medium for producing and delivering new services and collections. Because we cannot anticipate all of the potential services offered within the digital library, flexibility, openness, and extensibility are paramount criteria in design of the system. The UMDL addresses these criteria by providing infrastructure to coordinate the activities of the individual system modules, or *agents*, that perform the actual library services in the system.

In this paper, we first describe UMDL's agent-based infrastructure, its commerce framework, and the key role of commerce in the system. Second, we describe in further detail how the UMDL framework supports a wide range of negotiation mechanisms. Finally, we describe a simple scenario built in UMDL, illustrating one pattern of negotiation supported by the system. The scenario is intended to be representative (albeit not necessarily realistic in detail) of the sorts of interactions prevalent in a digital library, and thus serves as a proof-of-concept for the underlying approach.

UMDL Infrastructure

The UMDL architecture realizes a model of the digital library intermediate between a completely open environment (e.g., the world-wide web), and an entirely self-contained system (e.g., proprietary services that impose central control over collection access). Indeed, we can view UMDL as a digital library *intranet*, not in the physical sense of operating on a LAN or WAN, but in the organizational sense of providing a *information services network* which furthers an organization's goals and policies. The openness of the system allows users to benefit from the wide variety of services and collections potentially available, while the controls of the architecture provide structure to manage choices and connections among the diverse modules participating in the system.

To work at this intermediate level, the UMDL infrastructure mediates interactions among *information agents*. The main responsibility of this infrastructure is to facilitate the performance of services by some agents on behalf of others. This facilitation includes disseminating information about the availability and demand for services, and arbitrating connections among the providers and recipients of services.

Openness alone is not enough to gain participation by potential providers of information services, especially when interacting with the library infrastructure incurs some overhead. External agent developers (as well as internal library users, for that matter) will not participate unless they can identify some benefit not already available from the broader internet environment. Such benefits include the facilitation of connections mentioned above, but can also include more direct compensation, brokered through the system, acquired in exchange for the services or resources delivered through the digital library. Thus, the role of the UMDL infrastructure extends to determining what services are performed and under what terms, and in some cases monitoring and enforcing these terms.

Within the context of such an infrastructure, we can also address internal digital library objectives, such as allocating user, library, or network resources efficiently. For example, one type of library resource that may need to be internally allocated is access to site-licensed information. While a university site license may entitle all Michigan students to access certain information, only a limited number will be able to do so at one time.

Market-Based Resource Allocation

In a world of unlimited resources, users could avail themselves of all potentially beneficial services at their maximal levels of quality. However, in any real digital library, resources—including raw computational resources, human attention, access to intellectual property—ultimately have some cost.[1] Indeed, in even a moderately scoped digital library, there is potentially unbounded demand for computational resources. For example, any amount of preprocessing of data in the collections—such as indexing, meta-data gathering, or caching—might improve the response of the system to subsequent user requests. Given only finite resources, however, we cannot take advantage of all such opportunities. We also cannot generally try every method for accomplishing a given task, but rather must choose among those available based on resource requirements and prospects for success. We therefore seek principled methods for expressing operating tradeoffs and allocating computational resources toward their maximal expected benefit. Moreover, we require that these methods be sufficiently flexible to adapt to patterns of usage that evolve during the operation of the digital library.

In approaching this resource allocation problem, we treat the alternative information services as competing economic activities. Given a measure of priorities over the end-user services provided, the various agents effectively compete to provide the highest level of service using the minimal computational resources. One central capability of agents is thus to be able to reach agreements on suitable compensation. The goal is to achieve an efficient overall allocation of resources towards the optimal provision of services to users.

To organize the processing activities within an economic framework, we view interactions between the agents as supplier-producer relationships, where each agent produces value-added information products from the input products provided by others. Agents dynamically connect with each other as opportunities arise for mutually beneficial exchanges. The collections, represented by *collection interface agents* (CIAs), provide some ultimate "raw materials" in this process. Library end users, represented by *user interface agents* (UIAs) are the ultimate consumers of the "finished goods". *Mediator* agents ("middlemen") bridge the gap by bringing to bear knowledge, processing, storage, or other computational resources to improve in some way the expected value of the information as it passes along the chain from agent to agent.

The particular set of goods and the mechanisms by which they are negotiated and exchanged in the UMDL constitute the system's *market configuration*. In a dynamic and open environment, such as UMDL, it will be impossible to fix the configuration in advance. Moreover, finding an appropriate mechanism for determining exchange terms may vary depending on the current configuration, the good to be exchanged as well as other contextual factors. Therefore, to provide sufficient flexibility, UMDL must support dynamic market configurations through a extensible means of describing goods and wide variety of negotiation and payment mechanisms.

UMDL Negotiation Mechanisms

A negotiation and exchange facility for information goods and services must potentially support a wide variety of market types. For example, if one were to sell an electronic version of a magazine, it could potentially be sold in many different ways, such as:

- by subscription, individual issue, or individual article,
- to individuals, libraries, or other defined group,
- prior to paper publication, immediate on publication, or with delay,
- with or without license to further distribute.

The magazine might also be bundled with related services such as searching or notification of new articles.

Each of these different magazine goods might be offered under distinct sorts of negotiation procedures.

[1] Costs of some resources reflect the denial of opportunity to deploy them in alternate activities within the library. Costs of others reflect those incurred outside the library system.

For example, a single issue of the magazine might be offered by the supplier for a set price, as at a typical newsstand. The negotiation in this case degenerates to a take-it-or-leave-it proposition. On the other hand, the cost of search services might depend upon current congestion levels in the digital library system—when the system is more heavily loaded, a more timely response would cost more. In this case, the user may wish to specify a variable willingness to pay based on the immediacy of response. The system could then match the buyer's demand profile against current congestion profiles to determine an appropriate level of service.

Similarly, even the same magazine good could be exchanged via different negotiation mechanisms in different contexts. For example, a back issue of a magazine would typically be bought from the publisher for a set price, whereas one might haggle for it at a flea market. Thus it is the combination of the good and the negotiation mechanism which determines any market.

Each of the different good descriptions, negotiation and payment procedures may have advantages and disadvantages for the various types of information goods and services. While analyzing these issues is an important topic for research in digital libraries, we do not believe it appropriate for UMDL to dictate a particular solution. Instead, we provide generic facilities for defining these properties and rely on context and policy decisions to determine when and where they should be used.

The remainder of this section focuses on describing our generic negotiation specification approach. We use auction protocols as a framework to describe various mediated negotiations. However, there is no reason to believe that this approach would not extend to other types of negotiation methods as well.

Section presents a very simple, but representative, scenario—implemented in UMDL—where typical digital library goods and services are exchanged using one particular class of market mechanisms.

Market Facilitators

Within a multiagent environment, system agents specializing in brokering connections among other agents are often called *facilitators* (Genesereth & Ketchpel 1994). In the UMDL, *market facilitators*, or *auctions*, serve to determine which agents provide services to which others, under what terms. UMDL auctions operate by collecting offers and determining agreements consistent with those offers. For example, one simple kind of auction collects bids and settles them by finding a single price that clears the market, while others may perform a complicated matching procedure.

The negotiation protocol we have defined in the first UMDL prototype is based on a simple kind of market facilitator. There is one auction agent for each *good*, where a good may denote, for example, the delivery of digital objects, the provision of translation services, or other library product. Each auction accepts bids from agents interested in buying or selling that good. Bids include a demand schedule specifying the amount (generally, some quantity or quality measure) the agent is willing to buy or sell at various prices. The auction's task is to find a price such that supply balances demand, that is, the expressed amount agents are willing to sell in aggregate matches the amount other agents are willing to buy.

During the bidding process, market facilitators may also notify bidders of tentative prices, called *price quotes*, for use in comparison shopping or coordinating other activities. After a market clears, the auction may invoke some transaction execution services, depending on the nature of the exchange being mediated. For instance, we are currently planning to integrate UMDL with the NetBill payment mechanism (Sirbu & Tygar 1995) to exchange monetary fees for digital objects. As with negotiation mechanisms, the ideal payment mechanism may depend on many context-specific factors (MacKie-Mason & White 1996), and so it may be advantageous to support a generic payment interface (Ketchpel *et al.* 1996).

Specifying Auctions

We define auctions in UMDL via a straightforward auction specification language. This language serves two purposes. One is to organize the software implementation, so that components realizing particular auction features can be combined and reused. The second is that agents can refer to auction descriptions within the UMDL agent communication protocol to create particular auctions or to inquire about their characteristics.

In designing this specification language, we have attempted to exploit regularities common across different auctions. Building on other auction classification schemes (Engelbrecht-Wiggans, Shubik, & Stark 1983; Friedman & Rust 1993; Kagel & Roth 1995), we culled a working set of parameters that allow us to describe any of the major auction types.[2] See Table 1 for a partial set of these parameters and

[2] This effort was performed in conjunction with the Michigan Internet AuctionBot project, with assistance from Peter Wurman and William Walsh, (`http://auction.eecs.umich.edu/`).

their description. Recall that we do not impose a single auction mechanism in UMDL, instead supporting a variety of types, including those found in various survey articles (McAfee & McMillan 1987; Milgrom 1989; Smith 1989).

The central defining parameter of an auction is its clearing policy, which specifies the rules for determining the prices and allocation resulting from a set of bids. The economic properties of the auction—optimal bidding strategies, efficiency, welfare distribution—typically depend critically on these rules. For example, if a single unit of a good is being auctioned and incentive compatibility for buyers (i.e., a buyer's optimal strategy is to bid the true valuation of the item) is paramount, then a second-price (Vickrey) auction is appropriate. If there are multiple units and price discrimination (i.e., charging different buyers different prices) is acceptable, then the generalized Vickrey procedure described by Varian (Varian 1995) might be a good choice. Under alternate circumstances, yet other rules would yield the best balance of desiderata. For example, some key criteria for choosing auction types within UMDL are fostering market configurations that are incentive compatible and individually rational. The incentive compatibility criterion means that the protocol encourages rational agents to report their true values in bids. Individual rationality dictates that outside agents (and inside users) gain some advantage to participating in the system.

In our model, auctions can operate continually, clearing multiple times before they finally close, if ever. For example, one might specify that an auction should have a *clearing interval* of every three seconds, while the *final clear* should not occur until January 1, 2000. To get a better idea of how this works, consider the common English auction, where an auctioneer incrementally raises the price until there are no more takers. Such an auction would be specified in terms of our parameters as in Table 2.

Another kind of English auction, commonly seen on the World-Wide Web,[3] does not set the price quote according to this fixed-increment scheme. Instead, participants bid asynchronously, and the current highest offer is used as the price quote. This kind of English auction can easily be specified by changing the *price quote policy* in Table 2 to *first price*, (i.e., price quote equals current highest buyer's offer).

It is possible to specify unnamed auctions by filling in parameters in various ways. Of course, certain combinations of parameters would not be valid or might not make sense (such as having a price quote interval longer than the final clearing time). Such validity constraints must be enforced by the UMDL process that creates auctions from these parametric descriptions.

Creating and Finding an Auction

UMDL auctions may be created for any of several reasons. Shortages in basic computational resources (bandwidth, storage, processing) may cause the UMDL system to create auctions in those resources. Or, new markets may spring up in response to topical interests, for example, during hurricane season there may be a special market for weather services that track hurricanes.

One of the more common events triggering a new auction is when an information provider joins UMDL and wants to sell its product. To start the auction, the provider must first describe its product using UMDL's *goods description language*. The advantage of using a structured description language for goods is that agents may participate in multiple, related auctions covering the goods they are interested in buying or selling. For example, suppose the information provider, Sports-R-Us, is selling a magazine called *Teen Sports*. Based on specified features of this magazine, it might be determined automatically that there are several available auctions for exchanging this good, such as:

1. Teen Sports auctions *Teen Sports* magazine only
2. Sports-R-Us Magazines auctions all Sports-R-Us magazines
3. Teen Magazines auctions all kinds of teen magazines
4. Sports Magazines auctions all kinds of sports magazines
5. Magazines auctions all kinds of magazines

Sports-R-Us might choose any or all of these auctions to participate in. While they would probably create the Teen Sports and Sports-R-Us Magazines auctions, the others may be created by arbitrary agents. In deciding whether to join these auctions, Sports-R-Us would want to examine how they operated, by inspecting their specifications.

Suppose Sports-R-Us decides to participate only in a Teen Sports Auction. Since this auction does not exist already, they will need to create it by specifying the desired auction parameters. In this case, they would probably specify that this auction allows

[3] For pointers to auction-related web material, see `http://auction.eecs.umich.edu/other-auctions.html`.

Parameter	*Description*
Good Divisibility	Are good units discrete or continuous?
# Buyers	Number of buyers (one or many)
# Sellers	Number of sellers (one or many)
Information Revelation	Amount and types of information publicly available
Price Quote Policy	Method for calculating the price(s) reported in price quotes
Price Quote Interval	Interval (frequency) with which price quotes are posted
Clearing Policy	Method for calculating the resulting prices and allocations
Clearing Interval	Schedule (frequency) for clearing the auction
Final Clear	Time auction is terminated
Tie Breaking	Criteria for resolving tied bids

Table 1: Auction specification parameters.

only one seller, themselves. Once they have specified the auction type, an appropriate market facilitator agent, can be instantiated. The new auction advertises its services with the UMDL *Registrar*, a database agent that acts as an initial contact point for finding other agents in UMDL. [4]

On the other side of the economy, end users wishing to purchase *Teen Sports* interact with UMDL through their UIAs. UIAs may in turn contact specialized intermediary or broker agents that can help locate, recommend, or purchase products. However, in the simplest case, a UIA can locate the appropriate auction by querying the Registrar, receiving back pointers to auctions where that good can be bought. This process is depicted in Figure 1.

Auction Negotiation

Once an agent has located an appropriate auction, it can make an *offer* to buy or sell that good. The exact form of the offer depends both on the kind of auction as well as the preferences of the bidder agent. Table 3 shows an example of two offers to two different types of auctions (for different kinds of goods). The English auction is as specified above; the *Walrasian* auction is designed for multilateral exchange of continuous commodities, as described in Table 4.

An offer needs to specify both the price offered to buy or sell the good, and any qualifying information, such as expiration date. In the case of the Walrasian auction, the price component is variable, allowing agents to associate different prices with different quantities or quality levels. The *bid schedule* specified in the offer is essentially the agent's demand curve for the good. Offers may also include requests for notification of various auction events, such as new price quotes or when a bid expires.

The auction's clearing policy determines the allocation of goods and terms of exchange as a function of the offers it has received. For English auctions, the winner is the highest bidder at the last bid price. For the Walrasian auction, the clearing price is determined to balance buy and sell quantities, with each agent exchanging according to their expressed demand at that clearing price.

UMDL assumes all offers to be binding, subject to the commitment rules of the auction type. We do not currently support conditional bids, where bidders make offers contingent on sale or purchase of other goods. Instead, we deem it the bidders' responsibility to manage its interdependent bids by monitoring price quotes for its active auctions. Thus when an auction clears, any offers that have not expired and have not been retracted are considered valid offers. Pending integration of digital payment mechanisms, the only remaining task of the auction is to notify the two parties. In subsequent development, we expect the auction to initiate transactions using payment schemes agreeable to the parties involved.

[4] In practice, an agent may need to contact a separate *Auction Manager* rather than the Registrar. However, the two can be thought of as components of a single UMDL registry function.

Parameter	*Values*	*Explanation*
Good Divisibility	no	good exchanged as a single unit
# Buyers/# Sellers	many/one	
Information Revelation	bids transactions winner	most information about the process is publicly available, including bids, final clearing price, and winner identity
Price Quote Policy	+delta prices	auctioneer incrementally raises price
Price Quote Interval	immediate	price quote announced whenever it changes
Clearing Policy	first price	clearing price equals highest buyer's offer
Clearing Interval	inactivity	auction clears after idle period
Final Clear	inactivity	same as above
Tie Breaking	arrival time	earliest bid wins

Table 2: English auction specification: Price raised by auctioneer.

Figure 1: Finding an auction.

Scenario

To test and illustrate our negotiation mechanisms, we developed a simple market configuration reflecting a typical set of exchanges that might be expected in a digital library. We start with a UIA desiring to purchase various digital library goods and services, depending on its needs, budget, and prices. At the other end, a CIA wants to sell access to its collections, and acquire the system resources necessary to provide this service in UMDL. The "system resources" we consider here are really an amalgam of network and computational resources, which are bundled as a single good to simplify the initial scenario.

In the particular scenario we have constructed, depicted in Figure 2, two types of mediator agents can produce services that enhance the value of the CIA's product to the UIA. In particular, a *Notify* agent provides notification services, sending messages to the user when designated new material has been placed in a collection. A *Search* agent supplies search facilities over one or more collections. Both agents require some raw information resources, collection access and system resources, to produce their value-added services. As shown in Figure 2, the agents negotiate exchanges of relevant goods through their respective auctions, where arrows indicate the flow of goods.

The two mediator producers compete for demand of the UIA, and to acquire available resources. The UIA would likely consider the two services partly substitutable, as some amount of search can replace the service of the Notify agent, and vice versa. As the user's relative preference for notification compared to search increases, we would expect more of the system's resources devoted to the former rather than the latter.

Configuration

In general, configuring an information economy entails specifying the goods, the agents, and the market mechanisms. Since our focus was on market integration, we restricted the number and complexity of the goods and agents involved. As mentioned above, we abstracted all computational and network resources into a single "system resources" good. Similarly, we chose to measure other goods in one-dimensional quantity units. For example, we assume a scale for level of notification service, rather than support separate negotiation for the variety of parameterizations of the service (e.g., notify twice a day for three weeks of the event types).

Auction Type	*Offer Description*	
English	**Bid Schedule:**	buy 1 unit at \$5
many buyers	**Notify:**	price quotes
one seller	**Expiration:**	at next clearing
indivisible good		
Walrasian	**Bid Schedule:**	buy 5.3 units at \$1
many buyers		buy 1 unit at \$5
many sellers		sell 1.5 units at \$10
divisible good	**Bid Interpolation:**	linear interpolation between bid-schedule points
	Bid Extrapolation:	none
	Notify:	price quotes
	Expiration:	at next clearing

Table 3: Examples of offer formats for two auction types.

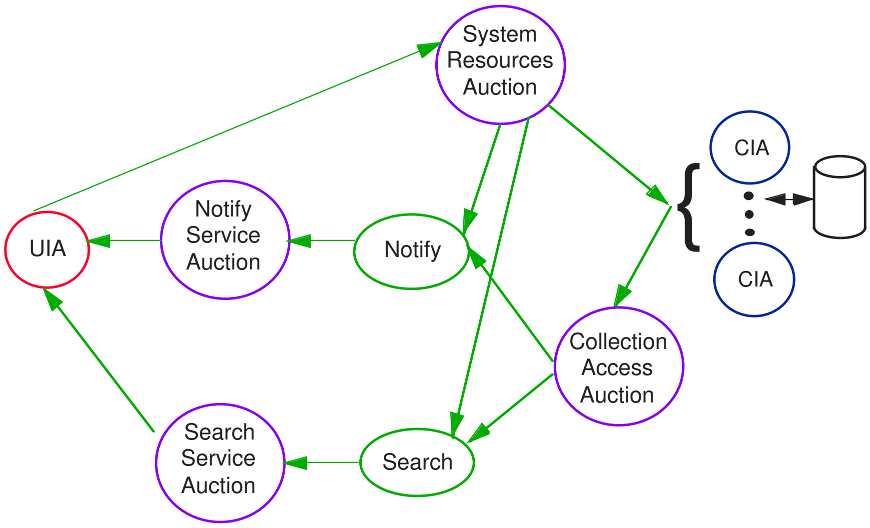

Figure 2: A simple UMDL negotiation scenario.

We designed our market configuration for the scenario within the general equilibrium framework employed in previous work (Mullen & Wellman 1995; Wellman 1993). In microeconomic theory, *general equilibrium* refers to the situation where multiple interconnected markets reach an overall price equilibrium. Clearing a single market in isolation is called *partial equilibrium*. Because a realistic digital library must simultaneously allocate numerous, highly interrelated resources, we deem general equilibrium the more salient analytical concept for this environment.

Adopting the formal general equilibrium framework, along with the competitive mechanism, resulted in three main advantages. First, we were able to use tools from economic theory to build the economy and verify the results. Second, the assumption of competitive behavior simplifies the design of agents. Competitive agents take prices as given—ignoring any market power they might possess, and thereby eliminating the need to reason strategically about other agents' bidding behavior. Moreover, competitive equilibria are guaranteed to be Pareto optimal. Third, only one kind of market mechanism, the Walrasian auction, is required.

Of course, the assumptions entailed by the general-equilibrium framework are also quite restrictive, and any analytical results that depend on these assumptions do not automatically transfer to the general UMDL context. Note however, that the UMDL negotiation mechanisms themselves are *not* inherently limited to competitive agents, Walrasian auctions, equilibrium-based transaction, or other simplifying features of the scenario presented here.

Consumers The general-equilibrium framework posits two basic types of agents, *consumers* and *producers*. The consumers in UMDL are the UIAs. Each UIA in the library is endowed with an allocation of system resources, which it can sell to the system in order to raise funds to purchase desirable services.[5] The consumer decides how much of each available service—in this case notify or search—to purchase depending on how much it values the service, the service's price, and how much it can afford. We represent user preferences by the CES (constant elasticity of substitution) utility function, which has the following form. The choice variables x_{sr}, x_n, and x_s, respectively, denote consumption of system resources, notification service, and search service.

$$U(x_{sr}, x_n, x_s) = (\alpha_{sr}x_{sr})^{\rho} + (\alpha_n x_n)^{\rho} + (\alpha_s x_s)^{\rho}. \quad (1)$$

The CES utility function provides analytic convenience and a reasonably straightforward way of expressing preferences between goods. The coefficients α_i weight the consumer's preference for good i, and the ρ parameter controls the degree of substitutability among goods. To run the model, we arbitrarily set $\rho = 1/2$ and decided that the user preferred the search service twice as much as the notify service, ($\alpha_s = 2\alpha_n$). The users also derived a small amount of utility from keeping system resources for a later time, so α_{sr} was set to a small value.

The consumer's demand function, $x(p)$, is derived by maximizing the above CES utility function subject to the budget constraint, given prices p. This problem has a closed-form solution, used by the UIA in generating offers for the different services desired. As shown in Table 3, the bid schedule for a UMDL Walrasian auction is specified as a series of price-quantity pairs, with linear interpolation to approximate the complete demand curve. By concentrating on points near the current price, a UIA can achieve reasonable approximation of the actual demand curve without having to specify very many points.

Producers Producers transform input goods into output goods according to constraints specified by their *technologies*. A competitive producer's goal is to maximize its profits, given its technology and the current relative prices of their input and output goods. In UMDL, we view all CIAs and mediators as producer agents.

For this scenario, we used three types of producers. CIAs use system resources to produce access to their collection. We assume the system is congested, so that the cost per unit of system resources provided increases with the load on the system. In other words, the more system resources are required, the more the congestion, the higher the unit cost. We represent this technology by a quadratic-cost production function where the production of collection access, y_{ca}, for a given consumption of system resources, x_{sr} is described.

$$x_{sr} = 0.01 * y_{ca}^2 + 0.05 * y_{ca}. \quad (2)$$

Another type of producer, Notify agents, bundle system resources and collection access to produce notification service. Buying one unit of system resources, x_{sr}, and one unit of collection access, x_{ca}, provides one unit of notification service, y_n.

$$y_n = min(x_{sr}, x_{ca}) \quad (3)$$

Finally, Search agents also bundle system resources and collection access, but produce search service rather than notification.

Walrasian Auctions Auctions determine the resource allocations and prices for each good. For this scenario, we only need to use one type of market mechanism, the Walrasian auction, described in Table 4. In particular, we set the *price quote interval* to be every two seconds and the *final clear* time to be two minutes from the starting time.

Offers, described in Table 3, can be submitted to the auction at any time up until the final clearing time. When an agent submits a new offer, that offer will supersede any previous offer. A price quote will be generated every two seconds by doing a hypothetical clear of all currently valid bids. This price quote is reported to each participating agent. Once an agent receives the new price, it may want to reconsider its bids for the other goods it is interested in and send out new offers. For example, if the price of search service changes, then an outstanding offer for notify service (based on how much the agent previously thought it would have to pay for search) will no longer be accurate.

Ideally, the Walrasian auctions clear until there is a simultaneous equilibrium across all of them, i.e., they reach a general equilibrium. One way of determining when the economy has reached a general equilibrium across related markets is when no bidding activity occurs in any of the markets. However,

[5] In the operational UMDL, we plan to endow the UIAs with currency directly, and supply basic computational resources through designated system agents. In the experimental versions of the system we employ a fake currency, which we call "BiblioBucks".

Parameter	*Values*
Good Divisibility	yes
#Buyers/#Sellers	many/many
Information Revelation	prices
Price Quote Policy	zero excess demand
Price Quote Interval	interval
Clearing Policy	zero excess demand
Final Clear	date & time
Tie Breaking	arrival time

Table 4: Walrasian Auction

this condition can not easily be assessed in decentralized, dynamic environments. Instead, each auction can be given a time limit before it will execute its final clearing. For our scenario, this time limit could be set long enough so that general equilibrium would be reached. Of course, in the general case, there are no guarantees that a general equilibrium will have been reached when the auction transacts. Depending on the nature of the market, this discrepancy might have negligible impact on the quality of the resource allocation or else it might require that different kinds of bidding protocols and periodic price-adjustment mechanisms be used instead.

At the auction's final clearing, the actual price is determined, as well as all buy/sell exchanges between agents. The terms of each transaction, the price and quantity to be exchanged, is sent to each of the matched agents.

Results

Within a digital library system, services will constantly be added and removed, reflecting newly available opportunities as well as changing user interests and priorities. Our scenario captures the dynamic flavor of digital library negotiation by running the market for some time with only one service provider, adding a second service while the negotiation is in progress. From the simulation, we see how the market adjusts to a new equilibrium reflecting the modified situation.

Initially, there is only one service—notify—available for the UIA to purchase. With no other choices available, the UIA will spend most of its endowment of system resources to acquire notify services. Since system resources is the numeraire good in this scenario, (i.e., its price is fixed at one), and it acts essentially like money.

As each agent bids at different auctions, prices converge to the competitive equilibrium prices and quantities shown in Figure 3. If the auctions were to clear at this point, the UIA would sell 610 units of its system resources to purchase 200 units of notify service. In order to produce those 200 units of notify service, the Notify agent would have bought 200 units of system resources and 200 units of collection access. Finally, in order to produce 200 units of collection access, the CIA would require 410 units of system resources, as dictated by its production technology.

However, before the final clearing time for the auctions, a Search agent is dynamically added to the economy. Since system resources and collection access can now be used for search services as well as notify services, the economy readjusts to a new Pareto optimal allocation reflecting the new uses for those resources, as can be seen in Figure 4. Normally prices, as well as allocations, would change, but since the two competing services have the same constant-returns technology in this scenario, prices remain unchanged.

Notice that it is the end user, or UIA, who drives how system resources and collection access will be reapportioned between the service providers. In our case, we designed the UIA to prefer search services twice as much as notify services, at a given price. Figure 4 does indeed verify that, given the same price for the two services, the UIA bought twice as much search service as notify service.

In fact, since this economy was designed within a general equilibrium framework, we were able to use the underlying economic theory to build the scenario and verify our results before ever running the scenario in UMDL. Given the producers' technologies, we applied standard microeconomic identities to derive utility function parameters and endowments consistent with the desired behavior (Mullen & Wellman 1995). Once the economy was specified, we verified the expected competitive equilibrium by simulating the configuration in the WALRAS market-oriented programming environment (Wellman 1993). As expected, the UMDL negotiation mechanism reproduces the result predicted by the theory.

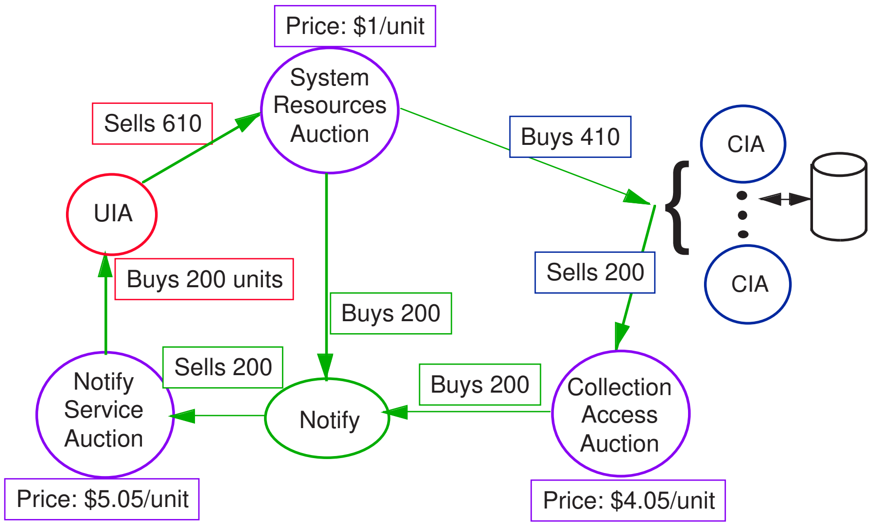

Figure 3: Finding initial equilibrium prices and allocations

Discussion

The preceding scenario illustrates the basic workings of negotiation within UMDL, and demonstrates the role of market-based negotiation in dynamic resource allocation. Negotiation mechanisms have thus far been implemented in only a subset of UMDL agents; we are working on providing generic negotiation facilities as part of the kernel agent software, and developing specific negotiation strategies for a broader range of agent types.

Market-based approaches have recently been investigated for a variety of problems in distributed resource allocation (Clearwater 1996). In particular, researchers at Columbia and IBM have explored over the past several years a variety of resource-allocation models applicable to distributed network resources (Ferguson *et al.* 1996), and potentially digital libraries. The SIGMA information retrieval system (Karakoulas & Ferguson 1995) is expressly conceived as a computational market, and we expect that the particular SIGMA information agents would be relatively at home in UMDL. Our own prior work studied a computational market model of network information services (Mullen & Wellman 1995), and served as a prototype for the UMDL information service economy discussed in this paper.

Several of the emerging large-scale digital library systems are incorporating payment mechanisms, primarily to support compensation for intellectual property usage. Most of these systems start from agreements negotiated out-of-band, or support a model of seller-specified take-it-or-leave-it pricing. The development of more flexible automated negotiation mechanisms is a relatively neglected problem, one that we are addressing directly in our work on UMDL.

Acknowledgments This work was supported by the NSF/ARPA/NASA Digital Library Initiative. Participants in the UMDL project have contributed to the concepts described here, especially William Birmingham, Edmund Durfee, Wendy Lougee, Jeffrey MacKie-Mason, and William Walsh. We are especially thankful to Steven Ketchpel, William Walsh, Peter Weinstein, and our anonymous referees for comments.

References

Atkins, D. E.; Birmingham, W. P.; Durfee, E. H.; Glover, E. J.; Mullen, T.; Rundensteiner, E. A.; Soloway, E.; Vidal, J. M.; Wallace, R.; and Wellman, M. P. 1996. Toward inquiry-based education through interacting software agents. *IEEE Computer* 29(5):69–76.

Clearwater, S. H., ed. 1996. *Market-Based Control: A Paradigm for Distributed Resource Allocation.* World Scientific.

Eatwell, J.; Milgate, M.; and Newman, P., eds. 1989. *Allocation, Information and Markets.* W. W. Norton and Company, Inc.

Engelbrecht-Wiggans, R.; Shubik, M.; and Stark, R. M., eds. 1983. *Auctions, Bidding and Contracting.* New York University Press.

Ferguson, D. F.; Nikolaou, C.; Sairamesh, J.; and Yemini, Y. 1996. Economic models for allocating resources in computer systems. In Clearwater (1996).

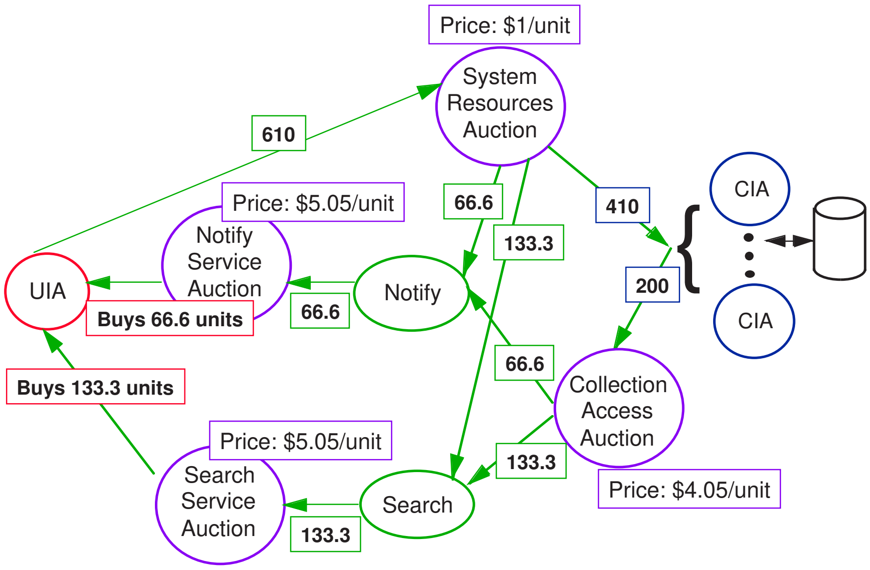

Figure 4: Finding final equilibrium prices and allocations.

Friedman, D., and Rust, J., eds. 1993. *The Double Auction Market*. Addison-Wesley.

Genesereth, M. R., and Ketchpel, S. P. 1994. Software agents. *Communications of the ACM* 37(7).

Kagel, J. H., and Roth, A. E., eds. 1995. *The Handbook of Experimental Economics*. Princeton University Press.

Karakoulas, G. J., and Ferguson, I. A. 1995. A computational market for information filtering in multi-dimensional spaces. In *AAAI Fall Symposium on AI Applications in Knowledge Navigation and Retrieval*.

Ketchpel, S. P.; Garcia-Molina, H.; Paepcke, A.; Hassan, S.; and Cousins, S. 1996. U-PAI: A universal payment application interface. In *Second USENIX Workshop on Electronic Commerce*.

MacKie-Mason, J. K., and White, K. 1996. An axiomatic approach to evaluating and selecting digital payment mechanisms. In *Telecommunications Policy Research Conference*.

McAfee, R. P., and McMillan, J. 1987. Auctions and bidding. *Journal of Economic Literature* 25:699–738.

Milgrom, P. 1989. Auctions and bidding: A primer. *Journal of Economic Perspectives* 3(3):3–22.

Mullen, T., and Wellman, M. P. 1995. A simple computational market for network information services. In *First International Conference on Multiagent Systems*, 283–289.

Sirbu, M., and Tygar, J. D. 1995. NetBill: An internet commerce system optimized for network delivered services. *IEEE Personal Communications* 2(4):34–39.

Smith, V. L. 1989. Auctions. In Eatwell et al. (1989).

Varian, H. R. 1995. Economic mechanism design for computerized agents. In *First USENIX Workshop on Electronic Commerce*.

Wellman, M. P. 1993. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research* 1:1–22.