In *Proceedings of the First International Conference on Multiagent Systems (ICMAS-95)*,
San Francisco, CA, USA, June 1995

A Simple Computational Market for Network Information Services

Tracy Mullen and **Michael P. Wellman**
Artificial Intelligence Laboratory
University of Michigan
1101 Beal Avenue
Ann Arbor, MI 48109-2110 USA
{mullen, wellman}@engin.umich.edu

Abstract

Visionary projections of a wide-area network teeming with intelligent agents describe an environment where end-users and their agents can pick and choose among a great variety of potentially valuable information services. However, neither network capabilities nor users' time and money are infinite. Computational markets provide one type of mechanism for allocating limited resources in such an environment in a distributed, dynamic way. Moreover, the underlying economic theory provides an analytical framework for predicting aggregate behavior and designing individual agents. In this paper, we describe a prototypical computational market model for information services distributed over a network. Our initial focus is on the economic problem of when and where to establish mirror sites for the more popular information services. Competitive agents choose to set up mirrors based on going prices for network bandwidth, computational resources, and the information service. Depending on the experimental setup, we observed a range of qualitative behaviors.

Introduction

Approaches to resource allocation in distributed systems can be bounded by two extremes. At one end (the "top down design" approach), we write a specification of the required behavior of the system in terms of software assets, architectural knowledge, and resource constraints. If our synthesized distributed system is guaranteed to satisfy these constraints, it is because we have built in a set of resource commitments that will lead to that result under the specified assumptions. At the other end of the spectrum (the "distributed agent" approach), rather than specify required behavior under given resources, we design protocols or mechanisms (Rosenschein & Zlotkin 1994) whereby a set of relatively autonomous software modules can, through interaction and run-time allocation of resources, achieve some desirable aggregate behavior using available system resources. Whereas this second approach is more flexible, it presents the problem of how we might determine *at design time* how well the various possible configurations of agents and available resources will achieve our desired results.

Market price systems constitute one well-studied class of mechanisms for allocating resources among distributed decision makers. By implementing a virtual market system for computational agents, we can hope to realize some of the desirable properties of markets in the distributed computing context. For example, in some well-defined circumstances, one can demonstrate that markets produce efficient allocations with minimal communication overhead. In that sense, we can sometimes reason about the mechanism in a principled way to decide whether it is appropriate. Indeed, one of the primary motivations of this *market-oriented programming* approach (Wellman 1993; 1995b) is to exploit the analytical framework of economic theory as a design tool for multiagent systems.

Market-Oriented Programming

The idea of market-oriented programming is to solve a distributed resource allocation problem by formulating a computational economy and finding its competitive equilibrium. To formulate a problem as a computational economy, we must cast the activities of interest in terms of production and consumption of goods, and define a set of agents that choose strategies for production and consumption based on their own capabilities and preferences and the going market prices.

To act in accord with the theory of competitive behavior, the agents must adhere to certain rationality conditions. *Consumer agents* are endowed with an initial quantity of goods and engage in trades so as to maximize their utility. *Producer agents* are associated with a *technology*, which specifies an ability to transform some goods into other goods. The sole objective of producers is to choose an activity within their technology so as to maximize profits. From the agents' perspective, the state of the world is completely described by the going prices; that is, the prices determine the maximizing behaviors. This arrangement is extremely modular, as agents need not expressly consider the preferences or capabilities of others, and communication consists exclusively of offers to exchange goods at various prices.

Since these computational economies are instances of general-equilibrium systems, the analytical tools and results of general equilibrium theory are directly applicable. In particular, under certain classical conditions, a simultaneous equilibrium of supply and demand across all of the goods is guaranteed to exist, be reachable via a distributed bidding process, and be Pareto optimal (that is, there is no solution that makes some agent better off without making some other one worse off). Other theoretical properties of equilibrium (to be illustrated below) can be used by designers to configure the system so that it achieves some desired aggregate behavior. Areas where market-oriented programming has been applied to date include transportation planning (Wellman 1993), distributed engineering design (Wellman 1995a), and allocation of computational resources (Bogan 1994; Doyle 1994). Although experience with the approach is still limited, some general understanding of the technique's characteristics is beginning to emerge (Wellman 1995b).

Some Related Work

Market or market-like mechanisms have been considered previously in Distributed AI research, most notably in the form of the Contract Net Protocol (Davis & Smith 1983). Although the contract net per se did not use real economic mechanisms, some economic concepts can be readily incorporated (Sandholm 1993). Recently, the market approach to resource allocation seems to be gaining in interest in the distributed computing community, for allocating computational resources of various kinds (Agorics, Inc. 1994; Clearwater *et al.* 1995; Ferguson *et al.* 1995; Harty & Cheriton 1995; Huberman & Hogg 1995; Kurose & Simha 1989; Waldspurger *et al.* 1992). Generally, these applications are centered around a global model for the resource, from which each agent or module calculates the marginal value of resource for itself. By using this value for bidding, the market allocates goods efficiently according to marginal value. An important distinguishing feature of market-oriented programming is that we are generally concerned with finding an allocation involving *multiple* interrelated resources. In other words, each agent is potentially interested in combinations or bundles of goods (resources, services), rather than a single type. For information networks of significant scope, such as digital libraries (Birmingham *et al.* 1994) or distributed databases (Stonebraker *et al.* 1994), it appears inevitable that considering the allocation of multiple resources and services at once will be necessary.

Network Information Services Economy

Current users of the internet are witnessing an explosive growth in the number and kind of information services offered. We expect this growth to continue and accelerate, and eventually for the range of services to start catering toward automated as well as human agents. Even in the current environment, economic issues are coming to the fore, as numerous proposals for internet billing protocols, transaction security, and electronic cash are being put forth for consideration. In some proposed schemes for internet pricing (MacKie-Mason & Varian 1994), prices and allocations are determined by dynamic bidding mechanisms. In such an environment, economically savvy computational agents will be at a premium.

It is widely recognized that the information services network of the future is fertile ground for computational agents. We believe moreover that the economic model of agent interaction provides a useful framework for system design. To explore this possibility, we are currently applying the ideas of market-oriented programming to one large-scale information services network, the University of Michigan Digital Library (Birmingham *et al.* 1994). Our design for this system is based on a network of specialized information agents, interacting as suppliers and producers in a virtual information-services economy (Birmingham *et al.* 1995). Within this system, efficient allocation of basic computational resources (memory, processing, bandwidth) as well as information goods and more complex value-added services will be a key to effective behavior of the overall system.

Blue-Skies Economy

For our initial exploration of resource allocation on an information services network, we have developed a simple computational economy consisting of one type of service offered at a variety of sites. The model is a direct extension of our previously developed transportation economy (Wellman 1993), which solved a distributed version of the multicommodity flow problem. The problem of routing information over a communications network is analogous; here, too, we have to choose routes simultaneously between several origins and destinations.

The economic issue we have focused on is the location decision for service provision: when and where should mirror sites be established? Our initial model represents a simple network composed of one internet site and two local sites and uses *Blue-Skies* (Samson, Hay, & Ferguson 1994), (a real-time interactive weather-images service at the University of Michigan), as our generic information service. In the model, Blue-Skies is produced on the internet and local users access it frequently. To continue accessing Blue-Skies from the internet means longer delays, but setting up a mirror site means using up local resources like disk space and computing, and also imposes an initial overhead cost of transporting the entire product line to the mirror site. Which setup makes more sense clearly depends on these costs, as well as current and anticipated access patterns by the end users on the local network. Figure 1 shows how one particular Blue-Skies economy was configured.

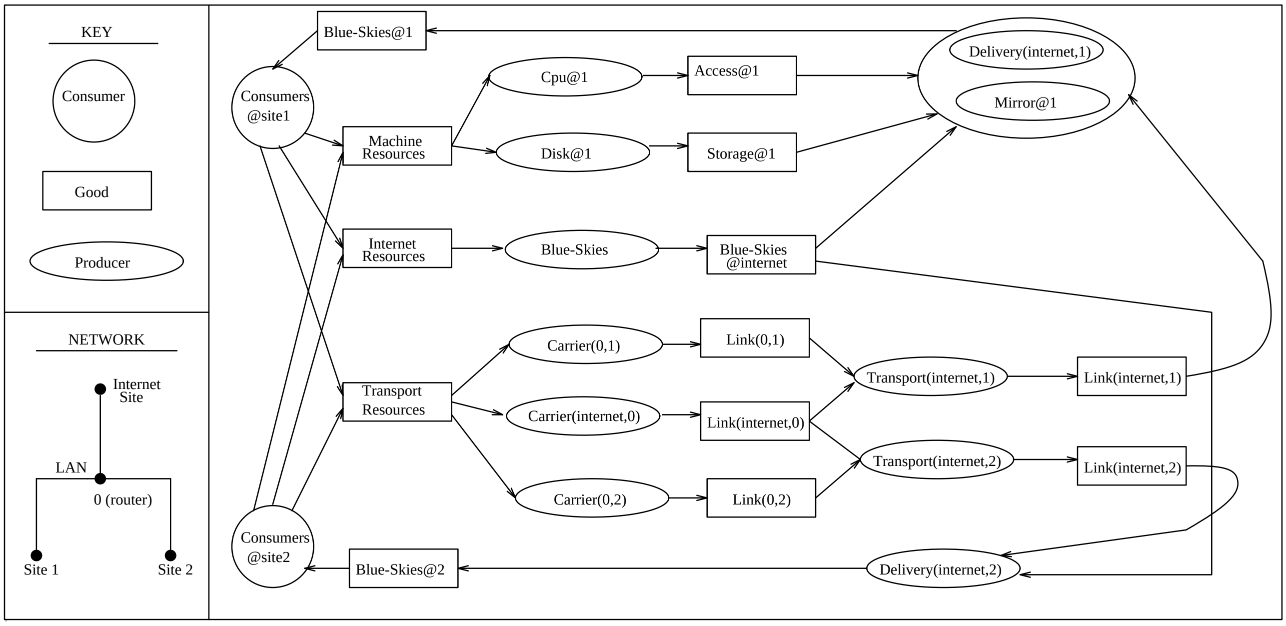

Figure 1: Blue-Skies Economy

Configuration

In order to design a computational economy, we need to specify the goods produced and consumed, the agents participating in the economy, and the agents' production and consumption behavior. Selecting the array of goods available strongly constrains the design space. The more standardized the goods, the simpler are the choices for each agent, but the less diverse the marketplace is. Thus, depending on design requirements, goods may either be decomposed according to various properties such as location, quality, and timeliness, or else they can be combined to hide relatively unimportant distinctions. In our model, we employed both of these options. For example, each of our consumers is endowed with basic *network resources* which they can "spend" buying Blue-Skies. Although we treat network resources as a single good, it actually denotes an amalgamation of network service characteristics (including throughput and reliability), which together constitute an "effective bandwidth". The units for this good combine both quantity and quality elements, which can be measured, for example, in terms of kilobytes of images delivered within a certain time period.[1]

On the other hand, location is an important characteristic here: users want to purchase Blue-Skies, not at the internet site, but on their local site. We represent this distinction by defining Blue-Skies as a separate good from Blue-Skies@local-site. (This is the most important way that this model extends the existing transportation economy.)

Producers are defined by their technology, which describes what resources are required to produce various amounts of output goods. In our model, we use five types of producers. *Carriers* produce transportation from one node to another, given network resources. We assume the network is congested, so that the cost per unit of effective bandwidth provided increases with the load on the communication link. Like Carriers, *manufacturers* of cpu access, disk storage, and Blue-Skies have decreasing returns to scale technologies. The third kind of producer, *transport arbitrageurs*, takes as input transport from node a to node b and transport from node b to node c, and produce as output transport from a to c. *Delivery arbitrageurs* are similar, except they bundle information services at a location (e.g., Blue-Skies@internet) with transport from that location to another (e.g, local-site), to produce that service at the second location (Blue-Skies@local-site). Finally, our fifth type of producer, the *mirror provider*, has the capability of transforming local storage and other resources into provision of the information service at its local site. It can also choose to access the service from another site instead of mirroring, in which case it acts just like the delivery arbitrageur. If it were possible to provide Blue-Skies using a mixture of sources (i.e., some from the net, some locally) then the two technologies encapsulated by the mirror provider could just as well be represented by distinct competing producers.

Consumers are defined by their initial endowment of resources and their preferences, which dictate the

[1] Since our entire model operates over the same time period, it was not necessary to explicitly determine whether the time period was per hour, per day, or per week.

Experiment	*Choices*	*Resale*	*Cache only*
No. 1	*site 1:* mirror, internet *site 2:* internet	internet internet	mirror internet
No. 2	*site 1:* mirror, internet *site 2:* mirror, internet	internet internet	mirror/internet internet/mirror
No. 3	*site 1:* mirror, internet *site 2:* site1, internet	mirror site1	oscillates oscillates

Table 1: Experiment Results

relative value to them of alternative combinations of information services and resource usage. For analytical convenience, we specify preferences using a CES (constant elasticity of substitution) utility function, specifying the tradeoff between network resources retained and consumption of the service good, Blue-Skies@local-site. Note that we can view a consumer agent in this model either as an individual end user, or as an aggregate of all the users at a particular site.

Behavior of the Model

In running our preliminary experiments, we have considered two different mirror site functions. First, the mirror serves as a cache for users at its local site. Second, depending on licensing issues, the mirror may also act as a reseller of the information service to other sites. We ran each of our experiments in two configurations. The first, labeled *resale*, permitted the mirrors to provide both the cache and resale functions. The second, *cache only*, prohibited resale beyond the local site. For each experiment, Table 1 lists the available choices for each site, as well as the the final market result for each configuration.

The economy configuration for Experiment 1 corresponds to that shown in Figure 1[2]. Site 1 must choose between being a mirror site or getting Blue-Skies from the internet (*site 1:* mirror, internet) while site 2 only had a single option—to get Blue-Skies from the internet (*site 2:* internet). In the case of the resale mirror, site 1 chose to get Blue-Skies from the internet. It had no one to resell the mirrored goods to, so that was not a profitable choice. For the cache-only setup, site 1 chose to have a mirror site; even though it couldn't resell the goods, it was still cheaper (given the cost parameters we used) to cache the product at its site than to perform all its access over the internet.

The results of all three experiments are what one would expect, except perhaps for the oscillation occurring in Experiment 3 for the caching mirror. The cycling occurs because setting up a mirror site reduces the overall traffic on the network. This in turn causes bandwidth prices to drop, which then makes it appear to the agents that it would be cheaper to get Blue-Skies via internet. Of course, as soon as they switch to getting the product from the internet, it drives the price back up, making the mirror appear more profitable. It is interesting to note, however, that the resale mirror model does not cycle. Site 1 choosing to be a mirror is only profitable if site 2 decides to get Blue-Skies from it, and once the system falls into this optimal choice, it will stay there.

Oscillation in the system can be attributed to two main sources: violation of competitive market assumptions and the presence of non-substitutable goods. For each cause, there are countermeasures available to the designer of network economies.

The competitive market assumption dictates that the agents be *price takers*, ignoring the effect of their own behavior on the resulting prices. This was clearly violated in our third experiment above, where the choice of one mirror producer dramatically influenced traffic and thereby moved prices. We expect that expanding the scale of the network economy, thus decreasing the influence of any one agent, would mitigate the effect on prices and tend to reduce the likelihood of the kind of synchronous flip-flop behavior seen in Experiment 3.

Non-substitutability of goods caused a problem in our original configuration, where we had separate goods for each kind of production resource: transportation, internet, and machine resources. Convergence was hindered by the fact that prices go to zero whenever there is an excess supply of some good. In our early tests, when site 1 chose to be mirror site, the economy was left with a large amount of transportation goods that could not be used elsewhere. Merging the three goods resolved that problem by making the different resources substitutable. However, the long-term solution here is also to scale up the economy. As long as there is *some* potential use for the excess transportation resources in the economy (even a very low-valued one), the problem of zero prices (and hence zero income for the consumers who own the resources) does

[2] We have pictured three separate basic resources in Figure 1 for clarity of understanding. In the actual experiments, they were combined into one "Resource" for reasons discussed below.

Blue-Skies: *Manufacturer*	
Demand Function:	$x = .01y^2 + .5y$
Historical Supply:	$y = 400$ units Blue-Skies@internet
Minimum Input:	1,800 units Network resources
Delivery(internet,site1): *Arbitrageur*	
Production Function:	$y = min(x_1, x_2)$
Demand Function:	$x = y$
Historical Supply:	$y = 200$ units BlueSkies@site1
Minimum Input:	200 units Blue-Skies@internet AND 200 units Link(internet,site1)

Table 2: Minimum Input Requirements

not arise.

In summary, we have defined a very simple computational market model for setting up mirror sites on an information services network. The model behaves predictably according to market principles, and exhibits a range of qualitatively distinct behaviors according to the market structure.

Designing a Blue-Skies Consumer

A consumer is defined by its initial endowment (which dictates its *budget*), and its preferences for different combinations of goods. If this information has not been specified externally, then it is up to the designers of the multiagent system to choose appropriate parameters. And as we have seen above, different configurations and parameters can lead to qualitatively different overall behaviors.

One of the benefits of the economic framework is that we can use the properties of competitive equilibrium derived from the underlying theory to design agents that will achieve the desired results. Although the following analysis is specific to the Blue-Skies economy, it demonstrates more generally the methodological utility of adopting a framework with firm theoretical underpinnings.

Based on our problem description, we made two assumptions. First, that the amount of Blue-Skies demanded historically (i.e., when local sites were supplied with Blue-Skies from the internet) was known, and that historic demand is an appropriate benchmark for demand in the expanded model with mirroring. Second, we assume that the CES utility function is a reasonable model of preferences. Given these two assumptions, our task is to set the parameters of the computational economy so that consumers can support their historic demands.

Finding the Initial Endowments

In any economy, producing a given amount of output requires some minimum amount of input be available. Since all of the basic resources are initially owned by the consumers, we have to make sure that the total endowment allocated to consumers is at least the minimum amount required to produce the desired aggregate output. Historical demand data tells us how much output was produced, so by working backwards through the production technologies we can derive the minimum amount of basic resource necessary. In Table 2, minimum input amounts are calculated for two particular producers. Similar calculations can be carried out for all the producers in the network.

Initial endowment amounts can then be determined in one of two ways, depending on the symmetry of the economic configuration. For symmetric economies, (where the network is symmetric and consumers have identical preferences), each consumer is assigned an equal share of the required input goods. Otherwise, one reasonable estimate of the required endowment is in proportion to the historical demands. Since our experimental economy is symmetric, we divide the total amount of network resources required (170,800 units) equally between our two consumers.

Utility Function Parameters

The CES utility function (for n goods) has the following form:

$$U(x_1, \ldots, x_n) = \sum_{i=1}^{n} (\alpha_i x_i)^\rho \quad (1)$$

The coefficients α_i weight the consumer's preference for each good, and the ρ parameter controls the substitutability between goods. For concreteness, we arbitrarily set $\rho = 1/2$ and seek a set of αs such that each consumer spends their network resources to buy Blue-Skies. Let $p = (p_1, \ldots, p_n)$ be the vector of prices for the n goods. The demand function x_i for good i, derived from maximizing the utility function subject to the budget constraint (total expenditures equal income) is:

$$x_i(p, income) = \frac{\alpha_i p_i^{-1/2} income}{\sum_{j=1}^{n} \alpha_j p_j^{1/2}}.$$

Our requirement is that the consumer's demands at equilibrium match the historical demand. Unfortunately, ensuring this is not as straightforward as it might seem; to evaluate the demand function we need to know the equilibrium prices, and the equilibrium prices depend on the demands. However, using the competitiveness assumption we can determine analytically the *relative* prices of the economy at equilibrium.

First, we use the fact that in equilibrium, price equals marginal cost (MC). If the cost $c(y)$ of producing y units of output is measured in units of input, then $MC(y) = p_{out}/p_{in}$, where p_{out} is the price of output, p_{in} the price of input, and $MC(y)$ the derivative of $c(y)$.

For example, the Blue-Skies producer has an input of network resources and an output of Blue-Skies@internet. Let the prices of these goods be p_{nr} and p_{bs}, respectively. Since the cost function for this producer is $c(y) = .01y^2 + .5y$, it follows that $MC(y) = .02y + .5$. Using the historical demand of $y = 400$, at equilibrium it must be the case that $MC(400) = .02(400) + .5 = p_{bs}/p_{nr}$. From this we can derive a constraint on the relative equilibrium prices, $p_{bs} = 8.5p_{nr}$.

By a similar analysis of the other producers' technologies, we can derive constraints on other combinations of relative prices, and then propagate these to express constraints between any connected pair of prices. For example, from the delivery arbitrageur we can determine that the price of Blue-Skies@site1 (p_{bs1}) equals the sum of prices of Blue-Skies@internet (p_{bs}) and transportation to site1, which (via some other constraints) comes out to $824.5p_{nr}$. Combining this with the constraint above yields $p_{bs1} = 833p_{nr}$.

We can also use the fact that for each consumer (in equilibrium), the ratio of its marginal utilities for pairs of goods equals the ratio of the prices for those goods. From this we can derive constraints on the α_i parameters of the consumer's CES utility function. We calculate marginal utility (MU) by taking the derivative of the CES utility function (1),

$$MU_i = \rho\alpha_i^{\rho} x_i^{\rho-1}.$$

Since $\rho < 1$, this means that marginal utility is undefined at $x_i = 0$. Therefore, even though the consumers really want network resources only for the purpose of trading them for Blue-Skies, we require that the consumers have some other direct use for the resource, not explicitly modeled by the economy. To ensure this, we arbitrarily set the consumption values for the resource goods to some small amount, for example, $x_{nr} = 16$. Solving for consumer@site1 (using $\rho = 1/2$), we have:

$$MU_{nr} = 1/2\alpha_{nr}^{1/2}(16)^{-1/2} = 1/8\alpha_{nr}^{1/2}$$

$$MU_{bs1} = 1/2\alpha_{bs1}^{1/2}(200)^{-1/2} = \sqrt{2}/40\alpha_{bs1}^{1/2}$$

From the analysis of producers above, we know that $p_{bs1} = 833p_{nr}$. Using the equilibrium condition $MU_i/MU_j = p_i/p_j$, we can find the relative values for the CES α parameters:

$$\begin{aligned} MU_{bs1}/MU_{nr} &= p_{bs1}/p_{nr} \\ (\sqrt{2}/5)(\alpha_{bs1}^{1/2}/\alpha_{nr}^{1/2}) &= (833p_{nr})/p_{nr} \\ \alpha_{bs1}^{1/2} &= (4165/\sqrt{2})\alpha_{nr}^{1/2} \\ \alpha_{bs1} &= (8.67 \times 10^6)\alpha_{nr} \end{aligned}$$

If we set $\alpha_{nr} = 1$, we will thus have one particular realization of a Blue-Skies consumer. And, since this is a symmetric economy, the consumer@site2 is the same, $\alpha_{bs1} = \alpha_{bs2}$.[3]

Note that in the foregoing analysis, we made several arbitrary choices ($\rho = 1/2$, $\alpha_{nr} = 1$, $x_{nr} = 16$). This is simply because the problem is underconstrained. To achieve a given behavior, there are in general many parameter settings that will do the job. If we had further behavioral constraints, these arbitrary settings are available as extra degrees of freedom.

Finally, we note that indeed, when run under WALRAS (our market-oriented programming environment, see (Wellman 1993)), this economy in fact produces the expected results. That is, the theoretical analysis is in fact validated by the behavior actually resulting from the distributed bidding behavior of the consumer agents we constructed.

Conclusion

Preliminary results show that under certain conditions the optimal mirror site configuration is chosen. However, because the current model is so small, problems with the competitiveness assumption and the limited range of resource-usage opportunities can cause the economy to oscillate. The solution to these problems is to scale up the economy by expanding the number of sites, adding more services, and distinguishing services based on different quality features, (e.g., having mirror sites that update less frequently but are cheaper,

[3] For non-symmetric economies, we may also need to calculate the profits of the producers, in order to use relationships depending on the income of consumers. For example, to find the Blue-Skies producer's profit (revenue minus cost):

$$\begin{aligned} p_{nr}y - c(y)p_{bs} &= 400p_{bs} - 1800p_{nr} \\ &= 400(8.5p_{nr}) - 1800p_{nr} \\ &= 1600p_{nr}. \end{aligned}$$

or more expensive sites carry a higher-quality version of Blue-Skies).

Other extensions include providing for explicit trading of licensing and distribution rights, and exchange of resources across multiple time periods. We expect that practical deployment of this model will also require compromises of the pure economic framework. For example, it would not be reasonable in general for the network of agents to reach a competitive equilibrium for each transaction, so different kinds of bidding protocols and periodic price-adjustment mechanisms will have to be explored.

Broader aims include developing general methodologies, and whenever possible, analytic techniques, for determining and directing the behavior of the system by specifying the initial economic configuration and the appropriate incentive mechanisms (possibly through taxes or subsidies). Although decentralized consumer and producer agents can act on these incentives in whatever manner they see fit individually, confidence in the aggregate behavior of the system can be maintained.

References

Agorics, Inc. 1994. Real-time video delivery with market-based resource allocation. Technical Report ADd004P, Agorics, Inc.

Birmingham, W. P.; Drabenstott, K. M.; Frost, C. O.; Warner, A. J.; and Willis, K. 1994. The University of Michigan Digital Library: This is not your father's library. In *Proceedings of Digital Libraries '94*, 53–60.

Birmingham, W. P.; Durfee, E. H.; Mullen, T.; and Wellman, M. P. 1995. The distributed agent architecture of the University of Michigan Digital Library (extended abstract). In *AAAI Spring Symposium on Information Gathering in Heterogeneous, Distributed Environments.*

Bogan, N. R. 1994. Economic allocation of computation time with computation markets. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science.

Clearwater, S. H.; Costanza, R.; Dixon, M.; and Schroeder, B. 1995. Saving energy using market-based control. In Clearwater (1995).

Clearwater, S. H., ed. 1995. *Market-Based Control: A Paradigm for Distributed Resource Allocation.* World Scientific.

Davis, R., and Smith, R. G. 1983. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence* 20:63–109.

Doyle, J. 1994. A reasoning economy for planning and replanning. In *Technical Papers of the ARPA Planning Initiative Workshop.*

Ferguson, D. F.; Nikolaou, C.; Sairamesh, J.; and Yemini, Y. 1995. Economic models for allocating resources in computer systems. In Clearwater (1995).

Harty, K., and Cheriton, D. 1995. A market approach to operating system memory allocation. In Clearwater (1995).

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

Kurose, J. F., and Simha, R. 1989. A microeconomic approach to optimal resource allocation in distributed computer systems. *IEEE Transactions on Computers* 38(5):705–717.

MacKie-Mason, J. K., and Varian, H. R. 1994. Pricing the internet. In Kahin, B., and Keller, J., eds., *Public Access to the Internet.* Prentice-Hall.

Rosenschein, J. S., and Zlotkin, G. 1994. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers.* MIT Press.

Samson, P. J.; Hay, K.; and Ferguson, J. 1994. Blueskies: Curriculum development for k-12 education. In *American Meterological Society Conference on Interactive Information and Processing Systems.*

Sandholm, T. 1993. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the National Conference on Artificial Intelligence*, 256–262. Washington, DC: AAAI.

Stonebraker, M.; Devine, R.; Kornacker, M.; Litwin, W.; Pfeffer, A.; Sah, A.; and Staelin, C. 1994. An economic paradigm for query processing and data migration in Mariposa. In *Third International Conf. on Parallel and Distributed Information Systems.*

Waldspurger, C. A.; Hogg, T.; Huberman, B. A.; Kephart, J. O.; and Stornetta, S. 1992. Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering* 18:103–117.

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.

Wellman, M. P. 1995a. A computational market model for distributed configuration design. *AI EDAM.*

Wellman, M. P. 1995b. Market-oriented programming: Some early lessons. In Clearwater (1995).