ELSEVIER

Robotics and Autonomous Systems 24 (1998) 115–125

Robotics and Autonomous Systems

Market-aware agents for a multiagent world

Michael P. Wellman *, Peter R. Wurman [1]

Artificial Intelligence Laboratory, University of Michigan, 1101 Beal Avenue, Ann Arbor, MI 48109-2110, USA

Received 9 October 1997

Abstract

A *computational market* is any collection of software agents interacting through a price system. Markets can provide effective allocation of resources for a variety of distributed environments, and economic analysis is a powerful design tool for interaction mechanisms. The spread of computational markets puts a premium on market-aware agents, and presents a case for market awareness on the part of agent developers and AI researchers as well. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Multi-agent systems; Computational markets

1. Introduction

By its very name, the workshop on "Modelling Autonomous Agents in a Multi-Agent World" (MAAMAW) [2] envisions a universe populated by numerous (presumably artificial) agents, acting and interacting autonomously. This interaction produces behaviors of complexity beyond our means to predict – hence the need for the modeling effort called for in MAAMAW's first "M". In the years since this conference was founded, research trends and technological developments have conspired to render this multiagent vision a commonplace, to the point of cliché. Any Internet user even vaguely following the popular press will naturally anticipate a future where autonomous agents roam the net, serving our needs, representing our interests, engaging other agents, and generally interacting on our behalf.

The role of AI research is in part to create this multiagent world, but also to help us understand and design for it no matter how it emerges. Of course, this ability to understand and design – *engineerability* – is enhanced to the degree that we can define a multiagent world with some regularity, and identify principles by which our agents operate and interact. Much of the research presented at the MAAMAW workshop can be viewed as efforts to develop such principles, or to characterize the properties of particular multiagent worlds.

1.1. Market awareness

The premise of this paper – and of most of our research over the past several years – is that one particularly useful model for constructing and analyzing multiagent worlds is that of *market price systems*. We argue that autonomous agents should be "market aware", not in the sense of following trends in the software business, but rather that they should be adept in interacting with and through market institutions. We

* Corresponding author. E-mail: wellman@umich.edu
[1] E-mail: pwurman@umich.edu

0921-8890/98/$ – see front matter
PII: S0921-8890(98)00026-8

present below both computational and economic reasons that organizing agent interactions through markets offers important advantages. However, even if one is not persuaded by these arguments, one might still expect markets to be prevalent in multiagent worlds, simply because the real-world commerce system offers a "default interface" for artificial agents. Market rules tend to be generic and globally standard, providing an applicable way to interact for agents who have not prearranged some other approach. The ubiquity of market interactions among real-world agents is itself sufficient basis to predict that market interactions will be common among artificial agents.

The impetus for market awareness naturally extends from the agents to their designers, and also to designers of multiagent systems. Construction of engineerable multiagent worlds requires that we understand the implications of alternative configurations, alternative interaction mechanisms, and alternative agent behaviors. To the extent that the agents interact through markets, our task is essentially one of economic analysis and design.

1.2. Resource-focused interactions

Without loss of generality, every decision made in a multiagent system is really about *resource allocation*. In making this statement, we take a very broad view of what constitutes a resource. Resources include physical materials (RAM, wires, airplanes), as well as more abstract ingredients of activity (time, space, attention, expertise). The distinguishing characteristic of the resources we consider is that they are *limited*. Choosing to do something entails an allocation of attention and other activity resources to that thing in lieu of others. Conversely, an allocation of resources defines the activities done and not done. Shared or multiple use of resources may be possible, but with some limitations in the extent or scope of the sharing. There are at least two reasons for taking this resource-oriented perspective:

(1) Much of what agents do can be described by resources involved in their various activities. The activities themselves are usefully defined by the resources they require, and the value or resources they generate. One must choose among two activities only if they are exclusive, i.e., if there is some resource (e.g., the actor's attention) required by both. In general, the resource-focused description lends structure to the decision problem by highlighting *tradeoffs* among alternate activities.

(2) A large share of significant multiagent interactions can be characterized in terms of resources transferred across the agents. Describing them in this way provides structure to the problem by simplifying the interfaces between agents.

One simple form of interaction – an *exchange* – is defined directly by specifying the resources transferred from each agent to the other. Other seemingly different forms of interaction, such as one agent performing a task for another, can be described as an exchange where accomplishment of the task is a resource generated by the first agent and transferred to the second. Even when the interaction does not involve an exchange, the salient point of connection often centers on a resource of mutual interest. For example, physical agents colliding is an instance of contention for a particular instance of space resource.

Once we have a characterization of agent activity and multiagent interactions in terms of resources, then specifying the configuration of resources devoted by the various agents to their respective activities (over time, in a dynamic system) determines the outcomes of the activities, and effectively summarizes the state of the overall system.

Activities can be usefully divided into two categories. Agents *consume* some resources, acquiring direct value from the resources consumed. Agents use other resources to *produce* new resources which they exchange for mutual advantage with other agents. The two types of activities define two conceptually distinct sectors of a multiagent system, as diagrammed in Fig. 1. Basic resources (e.g., agent effort and any raw materials initially held by the agents) flow from the consumption sector into the production sector. Productive activities transform the resources (perhaps in several stages) into finished goods, which can be directly enjoyed through employment in consumption activities.

A configuration specifying what resources are devoted to what activities is called an *allocation*. Determining a resource allocation can be viewed as the problem to be solved by the multiagent system. The two-sector perspective lends further structure to the problem by separating (1) what resources initially exist and what results are valued (defined by the

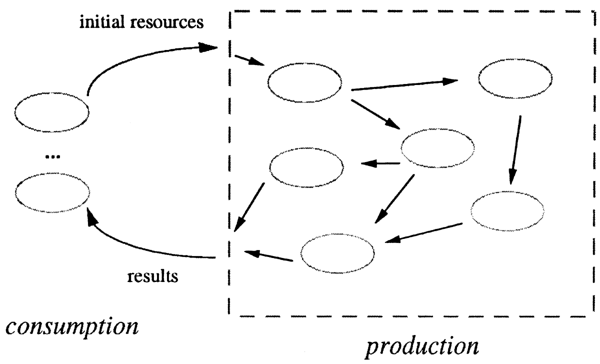

Fig. 1. Flow of resources in a multiagent system.

consumption sector), from (2) what can be done to produce results with resources (the production sector).

1.3. Mechanisms

The method by which an allocation is determined for a multiagent system is what we call a *multiagent interaction protocol*, or more simply, a *mechanism* [5]. A mechanism describes a communication protocol, in effect defining who can communicate with whom and what message types are allowed. The study of allocation mechanisms and their properties is known in economics as *mechanism design* [10,19].

Unmediated mechanisms involve bilateral, or multilateral, communication among all of the agents involved. Such mechanisms may not scale well to large numbers of agents. In *mediated* mechanisms, on the other hand, agents submit messages to some institution implementing the mechanism (or a part thereof). The process may be iterative, with the mechanism institution providing some feedback based on previous messages received. The process terminates under conditions prescribed by the mechanism rules, which then dictate the resulting allocation as a function of the pattern of messages.

Another conceptual benefit of the resource-allocation model is that it parametrizes the interface of agents to the rest of the system. The agents' behavior is divided into two realms. First, they participate in the resource-allocation mechanism to determine their available resources. Second, they apply these resources as they see fit. If the division is complete (i.e., the resources fully account for the multiagent interaction) then we have broken off a significant part of the problem – individual agents applying resources – that does not require interaction. To justify this separability, it must be the case that once we specify the allocation, what the agents do with their resources do not affect the others. In economic terminology, this is is to say that there are no *externalities*. Much of mechanism design is devoted to alleviating problems caused by externalities.

The task of a mechanism is to determine an allocation resulting from agent message-passing strategies. In designing a mechanism to solve allocation problems, we consider desirable properties for outcome allocations, as well as computational issues in determining allocations from agent interaction protocols.

A variety of desiderata for allocations may be considered. The most standard of these is *Pareto Optimality* (or *Efficiency*). An allocation is Pareto Optimal if and only if there does not exist another feasible allocation that is strictly preferred by one agent, and at least as preferred by all the rest. As designers of multiagent systems this is clearly the least we would want, as any inefficient allocation is dominated by another, in effect wasting resources. It is also the most we could require without introducing some judgment about the relative importance of satisfying various agents.

1.3.1. The sequel

In the remainder of this paper, we discuss some features of a particular class of resource-allocation mechanisms – that of market price systems. In Section 2 we motivate the introduction of markets by enumerating some benefits of structuring mechanisms according to a price system. We outline some of the design choices relevant to market systems in Section 3, and introduce some issues bearing on these choices. Our conclusion follows a few words on the design of market-aware agents.

2. Market price systems

There are a variety of mechanisms for solving resource-allocation problems, including such well-known examples as first-come-first-served and allocation by fiat. The defining feature of a *market-based* mechanism is that the allocation is mediated through a *price system*. In a market price system, we identify each distinct resource type with a *good*, and associate each good with a numeric specification of its exchange terms – the price. The basic rule of a price system is that goods may be exchanged according to the relative prices of the goods involved. For example if good j is assigned price p_j, then one unit of good 3 may be exchanged for p_3/p_7 units of good 7.

Thus, a price system imposes a constraint on the process for getting from one allocation to another. An allocation is *reachable* with respect to an initial allocation and a price system only if it is possible to reach the allocation from the initial through a series of exchanges at ratios determined by the given prices. Although the price system is restrictive (at least in the sense of narrowing the set of reachable allocations), basing a mechanism on prices has numerous advantages [12]. Here we mention several, without doing complete justice to all the underlying subtleties.

2.1. Prices define a common scale of resource value

Rather than consider all combinations of good exchanges, agents and mechanism institutions can simply relate each good to the common scale. This reduces the specification of values to a number of parameters linear in the number of goods, rather than in the size of the exchange space (which is exponential in the number of goods). The basic idea is that prices represent *marginal values*, which are often sufficient for evaluating incremental decisions.

2.2. Prices facilitate multilateral exchange via bilateral exchange

The common scale of values defined by a price system enables introduction of *currency*, tokens denominated in price units. This in turn enables complex transfers of multiple goods among multiple agents, without requiring multilateral communication or serial good transfers. That is, any multilateral exchange can be implemented as a set of bilateral exchanges, each of currency for a good, each from the original good's owner to its final owner. Such bilateral exchanges are generally much easier to arrange than would the effective multilateral transfer. Moreover, since most network environments support bilateral communication as a primitive, this property significantly enhances computational viability.

2.3. Prices summarize relevant information

An agent's appropriate behavior in a multiagent system depends in general on all other agents' preferences and capabilities, as all this may influence the relative values of resources. It is organizationally unrealistic to assume that agents know all this information about their counterparts (presumably one reason the system is decentralized), and computationally unrealistic even to ask them to contemplate the possibilities. Prices concisely represent the summary impact of other agents on good valuations, providing a compact specification of the agent's local decision problem. [2] If the agent's own influence on marginal valuation is negligible (i.e., the agent's action does not affect prices), then local behavior based on the summary information can support optimal (i.e., Pareto efficient) decentralization. Under similar conditions, an agent can determine the advisability of undertaking a newly available productive activity by evaluating its profitability with respect to given prices [22]. Ygge and Akkermans [30] present a concrete model that illustrates how a price system can effectively decentralize a distributed control problem.

2.4. Prices structure the mechanism protocol

Narrowing the objective to determining a price significantly focuses the mechanism design space (Section 3.3), and facilitates analysis and comparison of candidate mechanisms.

3. Market design space

Once we determine that a particular resource-allocation problem is amenable to market mechanisms, we still face many decisions regarding how

[2] This assertion can be rendered precise and demonstrated in particular settings. Specifically, Mount and Reiter [17] have shown that the competitive protocol (i.e., behaving according to given prices) minimizes the dimensionality of the message space required to determine an efficient allocation for convex problems. Jordan [11] has further shown that this protocol is uniquely minimal.

to configure the market. In this section, we describe some of the options open to the designer of computational markets. The space of possible markets – *marketspace* – can be characterized as the composition of two design spaces:

$$\text{marketspace} = \text{good space} \times \text{mechanism space}.$$

Good space comprises the ways in which we could specify the resources to be allocated in the system. The set of ways in which we might determine an allocation of those resources constitutes *mechanism space*. In the context of markets, we limit our attention to mechanisms based on prices. We consider both of these subspaces below, following a discussion of the overall goals of market design.

3.1. Design goals

Economists and game theorists have introduced a rich set of concepts useful for evaluating the properties of alternate allocation schemes. We present some of the more important ones below. A more detailed explanation of some of these can be found in Campbell's monograph on resource allocation [5, Chapter 2]. Rosenschein and Zlotkin [20, Chapter 2] discuss a similar set of desiderata in the context of multiagent systems. Note that these criteria apply to all resource allocation mechanisms, not just market systems.

- *Privacy preservation.* Informally, a mechanism preserves privacy if it does not require that individuals account for each others' private information in their own behavior. Privacy in general refers to the degree that information known specifically to individual agents is not revealed to the others. All else being equal, privacy reduces the ability of agents to gain strategic advantage by reasoning about or manipulating the beliefs of others.
- *Individual rationality.* A mechanism is individually rational if no agent would prefer not to participate in the mechanism at all (e.g., to take its initial situation as the result). The set of allocations such that no proper subset of agents can improve their lot by not participating is called the *core*.
- *Efficiency.* As mentioned above, Pareto efficiency is a natural criterion for evaluating the outcomes of resource allocation mechanisms.
- *Feasibility.* Even more primary than quality of the outcome is whether the allocation is feasible. An allocation is feasible if the resources assigned to activities are actually available, either as part of what initially exists or as a product of some productive activity assigned by the allocation. Feasibility may be difficult to ensure when mechanisms are distributed according to the resource. For example, if an agent's ability to produce X is contingent on its acquisition of Y, its negotiations for these resources are inherently coupled. Mechanisms that manage separate negotiations must provide some policy to cover unachievable commitments.
- *Incentive compatibility.* A mechanism is incentive compatible if an agent can do no better than by acting truthfully. This is a desirable feature because it allows the agent to consider only its own state, thereby simplifying its task. That is, incentive compatibility eliminates any advantage in speculating about other agents' strategies, and hence reduces the cognitive burden (information gathering and processing costs) for each agent. Moreover, if a mediator can extract all private information held by the agents through such a mechanism, it can determine an optimal allocation with respect to its other criteria.
- *Convergence and equilibrium.* Many mechanisms take an iterative form, on each iteration accepting messages from the agents and announcing a tentative allocation. Such a mechanism *converges* if it approaches a particular allocation over time. If an allocation is self-enforcing, in the sense that each agent would choose its part of the allocation given the information reported by the mechanism, we say that it is an *equilibrium*. For example, a *competitive equilibrium* is an equilibrium allocation in which each agent gets its optimal choice, *given* resource prices reported by the mechanism. Under certain conditions, competitive equilibria can be shown to exist [13], along with convergent mechanisms. Under fairly general circumstances, any such equilibrium must be Pareto efficient.

3.2. Good space

Selecting the commodities which will be traded is often the most difficult part of the design process. Sometimes a resource-allocation problem has an obvious breakdown into commodities. Other times, there are many ways to slice the resources into commodities, with no clearly superior treatment. In our experience developing a variety of computational market systems, we have found that decisions about good space typically drive an extensive chain of related design decisions [27].

3.2.1. Time and uncertainty

One very common set of issues arises when goods have temporal extent, such as systems that schedule time in a factory [1], or on a CPU. In such cases, the *time* that a resource is available is a defining attribute. Today's newspaper is different than next week's, and today's newspaper available today is different than today's available next week. Of course, next week's newspaper available today would be the most useful of all! To represent this variety, strictly speaking, we would need separate goods for each combination (t_1, t_2) of the newspaper dated t_1 available at t_2. Restricting attention to $t_1 \leqslant t_2$ eliminates only half the cases. A reasonable approximation might be to treat all instances $t_1 < t_2$ identically as "old news", thus reducing the required goods to a number linear in the time periods. A more extreme simplification, commonly used, would be to consider only a single time period (i.e., *now*, usually implicit), and address allocation of future resources when their times come. Intermediate approaches, perhaps the most common, may consider some particularly important future resources, neglecting the rest.

In the FreeWalk computational economy [29], for example, agents are endowed with raw network resources (bandwidth) every time period, and negotiate for multimedia service (measured in quality units, QoS) over time. Depending on the agent's current and anticipated needs, as well as current and anticipated prices, it may choose to make intertemporal tradeoffs between services used at various points in time. However, rather than maintain markets in QoS for each time period, we make only a binary distinction, between current and future. In each time slice, we run the two-period model to near equilibrium, allocate service according to the "current" allocations, and roll the horizon forward by using the "future" allocations to set the initial conditions for the next round. [3] Although this approximate model presumably leads to less efficient outcomes than would the full set of temporal markets, having even one available futures market enables some of the most beneficial intertemporal exchanges to be effected. Thus, expanding from one period to two can provide a large share of the potential improvement of the full temporal model, at a small fraction of the computational cost.

Another common issue is *uncertainty*, where agents must make decisions about deploying resources without full knowledge of the outcomes of their activities. In some cases, agents may be able to improve overall welfare by exchanging resources *contingent* on resolution of some uncertain events. Whereas it would usually be inconceivable to introduce goods corresponding to every resource in every state of nature, it generally suffices to introduce *securities* – one for each state – paying off in some designated resource unit if the state obtains [13]. Of course, this too may require too many goods, and so we will often prefer to cover only a selected set of important contingencies.

The idea that markets in uncertain propositions can be used to coordinate decentralized behavior under uncertainty was first proposed in the context of multiagent systems by Hanson [8]. In our own work, we have shown how securities markets can serve to aggregate the beliefs of multiple agents with possibly divergent information sources [18]. However, to date there has been relatively little investigation of the potential of contingent goods for more general problems of resource allocation under uncertainty in multiagent systems.

3.2.2. Example: good space for a factory scheduling problem

To illustrate the subtlety of good specification decisions, let us consider a (much simplified) factory scheduling problem in some more detail. Imagine a factory that can make several different products, for instance, one that makes different colored bucket seats for cars. This factory may supply seats to several different automobile companies. These companies need the seats at specific times in order to install them during automobile assembly. Depending upon the model they are building, they need different numbers of seats:

[3] It is not necessary to reach strict general equilibrium each period to ensure feasibility, as only the current period is actually deployed in productive and consumptive activities.

two for a sports car, four for a passenger car, and six for a minivan. The automobile company needs the complete set, on time, to produce the car.

A default method for solving this problem would have the companies place orders for all of the seats they need, and let the factory choose which to serve first. Any renegotiation would be handled bilaterally and via unrestricted protocols. If we instead wish to cast this in market-oriented terms, our task is to design the market structure. There are at least two very different ways to formulate the good space for this problem.

One formulation adopts seat bundles as the primary good. If a company wants to build a red minivan, it will offer to buy a bundle of six red seats from the factory. Since red seats delivered at 5:00 today are not interchangeable with blue seats or red seats delivered tomorrow, we would have to have one market in every combination of seat color and delivery time.

A second approach would treat factory time as the primary good. Automobile companies would buy time at the factory (perhaps in units of time sufficient to produce a car seat), and decide for themselves which color seats to produce.

An advantage of the second approach is that the factory need not make decisions about what color seats to make. A disadvantage is that the automobile companies would need to know what factory resources go into making a seat. In general, design decisions about good space have implications for information burdens on participating agents, and on what scope of markets they need to consider. In a general market for factory time, the factories can potentially provide their service for a variety of purposes by consulting a single market. In a general market for car seats, the automobile companies can potentially obtain seats produced by a variety of methods (e.g., different factory types) without needing to shop across factory markets.

In this example there is no clear basis for deciding which formulation is superior. Actually, we do not necessarily have to make a choice – we can create markets of both types and let agents translate between them. [4] But whether we decide on either or both, the choice of good formulation may have a significant influence on the possible mechanisms for determining the price and allocation of these resources.

3.3. *Mechanism space*

Without much loss of generality, we identify the problem of designing a mechanism for determining prices with that of designing an *auction*. The general definition of an auction (see [14]) is quite general indeed. It essentially includes any well-defined mediated protocol that determines prices as a function of messages submitted by participating agents.

As a type of protocol, an auction is defined in terms of the messages that comprise it. In their full generality, it is not possible to define required elements of an auction protocol. Nevertheless, most actual and proposed auctions include at least the following component message types.

- *Bids* typically represent an agent's willingness to engage in some exchange or set of exchanges. For example, a bid might specify a demand function, indicating the agents' desired quantity of a resource as a function of price.
- *Price quotes* convey to agents information about the state of an auction in the interim before final price determination and allocation.
- *Notifications* inform agents of the results of an auction, i.e., a price-based allocation. Typically, the auction reports to agents the terms of contracts they have agreed to.

A full specification of the variations on these message types and other protocol parameters is beyond the scope of this paper. We are exploring this space systematically with our configurable auction server, the Michigan Internet AuctionBot [28]. [5] In the following, we provide a brief elaboration of some issues in auction design, organized around the three primary message types.

3.3.1. *Bids*

Bids represent the agent's willingness to exchange, typically specifying quantity demanded of a good as a function of its price. For single-unit goods, this amounts to specifying a threshold price. An auction

[4] We call agents that implement identity relations *arbitrageurs*, and find them generally useful for smoothing the operation of computational markets [27], e.g. by mapping combinations of goods directly produced to notional goods directly consumed [26].

[5] `http://auction.eecs.umich.edu/`

might impose any of a variety of restrictions on the scope of bids, for example that they include a limited number of price-quantity pairs, that they be in discrete units, or that they have some regular form. In addition, the auction may restrict changes in bids, such as disallowing withdrawals or decreases in offer prices. Or, it may limit which agents are allowed to place buy or sell bids in the auction. As long as the rules are explicit and computable as a function of the joint bid state, the auction can enforce them simply by rejecting bids that violate the constraints.

3.3.2. Price quotes

Price quotes are the means by which the auction disseminates information about the state of the market. We call them "price quotes" because they often take the form of hypothetical prices at which agents would or could exchange if the allocation were to be determined from the current state. To compute quotes interpretable in this way, the auction can usually just execute its allocation-determining, or *clearing*, algorithm in a hypothetical mode. The quote may answer such questions as "what would the price be if the auction cleared now?", or "what would I have to offer to obtain a unit of the good, given the state of the other agents' bids?" The answer to this latter question is generally called an *ask quote*. A *bid quote* represents the price at which one would have to offer to sell in order to transact in the current state. The terminology comes from the "bid/ask spread" commonly used to describe the state of bidding for commodities in organized exchanges.

A chronology of price quotes represents an historical description of the state of an auction at a series of particular times. Auctions may report other historical information, such as the terms of actual previous transactions (assuming that the auction can be associated with a resource *type* with some meaningful history). By reporting such an information, auctions reduce the information inequality that would otherwise exist between agents participating in previous transactions and everyone else.

Exactly what interim information is revealed and when, can have a significant effect on agents' bidding strategies and the overall effects of auctions. Theoretical analysis of auctions [14,16] is always based on explicit assumptions about price quotes. For example, *sealed-bid* auctions are defined as those that eschew price quotes altogether.

Sometimes, limiting or even eliminating information revealed through price quotes can be a beneficial policy. If information about one's bid will be revealed – even indirectly – through price quotes, then one must consider the potential negative consequences of this revelation in constructing a bid. Similarly, one should also take into account whatever the price quote revealed about the other agents. Both considerations add to the cognitive burden on individual agents, and may lead to inefficient outcomes attributable to the agents' strategic behavior.

On the other hand, if multiple interdependent resources are auctioned at once, an agent's value for one good depends substantially on the prices at which it could buy or sell others. In that case, a price quote is essential for effective coordination of the agent's exchanges. For example, in the tatonnement and related protocols [6,13], agents adjust their bids for all goods based on changes in prices for all goods. Indeed, methods for reaching general equilibrium invariably rely on such iterative price adjustment processes.

One of the most difficult issues for auction designers is how to encourage agents to participate in the mechanism during this iterative search phase – thus making the search meaningful – despite their disincentive to reveal information about themselves. One useful approach is to impose *activity rules*, usually in the form of conditions for auction participation based on level of bidding activity. For example, in the FCC spectrum auctions, an agent was dropped out of the iterative bidding process whenever it failed to maintain some number of high bids among the recent bidding history [15].

3.3.3. Notifications

Notifications per se are not very interesting; they merely report the result of the auction process. What is salient, of course, is how the allocation is determined itself.

The allocation determination event is called a *clear*, as in clearing a market. The auction clears according to a preset schedule, which may specify particular times or bidding events that would trigger a clear. At one extreme – the continuous auction – every bid could trigger a potential clear, depending on whether it matches an existing bid. At another – the one-shot auction – there is only a single clear, typically at a fixed designated time.

The auction's clearing algorithm matches buyers and sellers and establishes a price at which they are willing to transact. When goods are divisible and bids take the form of continuous decreasing demand functions, a natural clearing algorithm (that of the *Walrasian* auction) would calculate the exact price at which buyer and seller demands balance. For indivisible goods, or other circumstances with discontinuous demands, such perfect clearing prices do not necessarily exist.

Uniform pricing algorithms establish a single price at which every transaction formed during that clear will execute. Examples of uniform pricing policies include the Mth and $(M+1)$st pricing rules, where M refers to the number of units offered for sale. Under these policies, the clearing price is the Mth (or $(M+1)$st) highest of all bids submitted – counting each unit, both buy and sell. When $M = 1$, these prices reduce to the well-known first- and second-price auctions (with reservation prices). As Vickrey [24] demonstrated in his seminal work on auction analysis, the second-price sealed-bid auction is incentive compatible with respect to bids to buy. The $(M+1)$st-price sealed-bid auction is similarly incentive compatible for buyers, and the Mth-price auction for sellers. It can be shown that no uniform-price auction can be simultaneously incentive compatible for both buyers and sellers.

Incentive compatibility might be one reason to adopt some form of *discriminatory* pricing, in which different agents may pay or receive different prices for the same good. Another (more fundamental) reason may be the potential nonexistence of an exact uniform price, as mentioned above. In some cases, discriminatory schemes may support more efficient overall results. However, they may also be difficult to enforce, and the perception of unfairness might also impede their acceptance unless due care is taken in design.

4. Agent design

Strictly speaking, agents are not part of marketspace, as designers of the market structure typically cannot control the participating agents. [6] Indeed, many of the design criteria enumerated in Section 3.1 specifically address this element of reality.

As market mechanisms become more prevalent (for reasons adduced above and otherwise), design of agents specializing in market interactions will become an important category of agent research. Artificial intelligence and economics both have much to say about the principles of agent design [3], and we expect that design principles specific to market mechanisms will build on these foundations. Although we make no attempt to survey existing work bearing on this topic, at least one current research topic seems worthy of mention.

One way to classify agents in a multiagent system is with respect to the degree and depth to which agents attempt to reason about each other. Vidal and Durfee's concept of a k-level agent [25] can be described inductively, roughly as follows:

- A 0-level agent does not reason directly about others.
- A 1-level agent forms models of other agents as 0-level agents.
- A k-level agent forms models of other agents as $(k-1)$-level agents.

Game-theoretic agents, in contrast, would typically be regarded as ∞-level, assuming common knowledge of the game and common knowledge of mutual rationality.

We find it useful to distinguish further among 0-level agents acting within a price system, classifying them as *competitive* or *strategic*, according to whether they take into account their own effects on prices. (An agent 1-level or above is strategic by definition.) This automatically induces distinctions at higher levels, based on what type of 0-level agent the hierarchy is founded on. We might also distinguish variant forms of k-level agents, according to whether they reason about other agents individually or in the aggregate.

In one recent investigation [9], we compared agents of different types acting within a simple market game. Specifically, the experiments included 0-level

[6] Cases where agents are under the control or partial control of the designers may call for very different approaches [4]. In such situations, the *agency* of the modules is more ambiguous, as their autonomy is somewhat compromised.

(competitive) agents, along with 1-level agents that formed an aggregate model of the others as competitive. We found that although the higher-level agents converged to "correct" models (i.e., the system reached an expectations equilibrium), they were still likely to be worse off than had they behaved as simple competitors. The reason is that the strategic agents may have had no way to verify their conjectures about actions not taken, and so they often missed favorable opportunities. In a related model, Sandholm and Ygge [21] also found that competitive agents could outperform their strategic counterparts under uncertain conditions, as the more sophisticated policies were more sensitive to errors in models formed.

Clearly, these types of experiments merely scratch the surface. We cannot realistically expect to obtain general results about the superiority of some broad class of strategic policies, as so much depends on the particulars. Indeed, within any particular "level" class, there are so many ways to realize reasonable bidding behaviors that we clearly need some further exploration and structuring of this space. Sierra et al. [23] specify a variety of bilateral negotiation strategies, and present theoretical and empirical evidence bearing on their convergence and performance.

5. Conclusion

The argument for market awareness elaborated here has two keystones:

(1) Agents interact largely through resources, and thus resource-allocation covers a significant part of our problem as designers of multiagent systems.

(2) Price systems provide structure to the allocation problem, and offer numerous advantages for decentralizing decisions across autonomous agents.

In exploring the rationale and implications for the market-based approach, we have identified many issues bearing on the design of computational markets. If we have convinced the reader that (1) the economic approach brings with it useful concepts for understanding and evaluating alternative designs, and (2) intellectually challenging and practically important problems remain for researchers in this area, then we have achieved our goal.

Acknowledgements

Our perspectives on computational markets have been enriched by ideas of many within the Decision Machines research group at the University of Michigan AI Laboratory, the works of John Cheng, Junling Hu, Tracy Mullen, David Pennock, and William Walsh playing particular roles. We have also benefited from discussions with Jeffrey MacKie-Mason and other members of the MARX project, and participants in the MAAMAW-97 workshop. This work has been supported in part by the Air Force Office of Scientific Research and the National Science Foundation.

References

[1] A.D. Baker, Metaphor or reality: A case study where agents bid with actual costs to schedule a factory, in: S. Clearwater (Ed.), Market-Based Control: A Paradigm for Distributed Resource Allocation, World Scientific, Singapore, 1995.

[2] M. Boman, W. Van de Velde (Eds.), Multi-Agent Rationality: Proceedings of the Eighth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-97), Lecture Notes in Artificial Intelligence, Springer, Berlin, 1997.

[3] C. Boutilier, Y. Shoham, M.P. Wellman, Economic principles of multi-agent systems (Editorial), Artificial Intelligence 94 (1–6) (1997) 1–2.

[4] R.I. Brafman, M. Tennenholtz, On partially controlled multi-agent systems, Journal of Artificial Intelligence Research 4 (1996) 477–507.

[5] D.E. Campbell, Resource Allocation Mechanisms, Cambridge University Press, Cambridge, 1987.

[6] J.Q. Cheng, M.P. Wellman, The WALRAS algorithm: A convergent distributed implementation of general-equilibrium outcomes, Computational Economics 12 (1) (1998) 1–24.

[7] S. Clearwater (Ed.), Market-Based Control: A Paradigm for Distributed Resource Allocation, World Scientific, Singapore, 1995.

[8] R. Hanson, Even adversarial agents should appear to agree, in: IJCAI-91 Workshop on Reasoning in Adversarial Domains, Sydney, Australia, 1991.

[9] J. Hu, M.P. Wellman, Self-fulfilling bias in multiagent learning, in: Proceedings of the Second International Conference on Multiagent Systems, Kyoto, Japan, 1996, pp. 118–125.

[10] L. Hurwicz, The design of resource allocation mechanism, in: K.J. Arrow, L. Hurwicz (Eds.), Studies in Resource Allocation Processes, Cambridge University Press, Cambridge, 1977, pp. 3–37; reprinted from American Economic Review Papers and Proceedings, 1973.

[11] J.S. Jordan, The competitive allocation process is informationally efficient uniquely, Journal of Economic Theory 28 (1982) 1–18.
[12] T.C. Koopmans, Uses of prices, in: Scientific Papers of Tjalling C. Koopmans, Springer, Berlin, 1970, pp. 243–257; originally published in the Proceedings of the Conference on Operations Research in Production and Inventory Control, 1954.
[13] A. Mas-Colell, M.D. Whinston, J.R. Green, Microeconomic Theory, Oxford University Press, New York, 1995.
[14] R.P. McAfee, J. McMillan, Auctions and bidding, Journal of Economic Literature 25 (1987) 699–738.
[15] R.P. McAfee, J. McMillan, Analyzing the airwaves auction, Journal of Economic Perspectives 10 (1) (1996) 159–175.
[16] P. Milgrom, Auctions and bidding: A primer, Journal of Economic Perspectives 3 (3) (1989) 3–22.
[17] K. Mount, S. Reiter, The information size of message spaces, Journal of Economic Theory 8 (1974) 161–191.
[18] D.M. Pennock, M.P. Wellman, Representing aggregate belief through the competitive equilibrium of a securities market, in: Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence, Providence, RI, 1997, pp. 392–400.
[19] S. Reiter, Information incentive and performance in the $(\text{new})^2$ welfare economics, in: S. Reiter (Ed.), Studies in Mathematical Economics, MAA Studies in Mathematics, 1986.
[20] J.S. Rosenschein, G. Zlotkin, Rules of Encounter: Designing Conventions for Automated Negotiation among Computers, MIT Press, Cambridge, MA, 1994.
[21] T. Sandholm, F. Ygge, On the gains and losses of speculation in equilibrium markets, in: Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-97), Nagoya, Japan, 1997, pp. 632–638.
[22] H.E. Scarf, The allocation of resources in the presence of indivisibilities, Journal of Economic Perspectives 8 (4) (1994) 111–128.
[23] C. Sierra, P. Faratin, N.R. Jennings, A service-oriented negotiation model between autonomous agents, in: M. Boman, W. Van de Velde (Eds.), Multi-Agent Rationality: Proceedings of the Eighth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-97), Lecture Notes in Artificial Intelligence, Springer, Berlin, 1997.
[24] W. Vickrey, Counterspeculation, auctions, and competitive sealed tenders, Journal of Finance 16 (1961) 8–37.
[25] J.M. Vidal, E.H. Durfee, Agents learning about agents: A framework and analysis, in: AAAI-97 Workshop on Learning in Multi-Agent Systems, July 1997.
[26] M.P. Wellman, A market-oriented programming environment and its application to distribute multicommodity flow problems, Journal of Artificial Intelligence Research 1 (1993) 1–22.
[27] M.P. Wellman, Market-oriented programming: Some early lessons, in: S. Clearwater (Ed.), Market-Based Control: A Paradigm for Distributed Resource Allocation, World Scientific, Singapore, 1995.
[28] P.R. Wurman, M.P. Wellman, W.E. Walsh, The Michigan Internet AuctionBot: A configurable auction server for human and softare agents, Second International Conference on Autonomous Agents, Minneapolis, MN, 1998, pp. 301–308.
[29] H. Yamaki, M.P. Wellman, T. Ishida, A market-based approach to allocating QoS for multimedia service, in: Proceedings of the Second International Conference on Multiagent Systems, Kyoto, Japan, 1996, pp. 385–392.
[30] F. Ygge, H. Akkermans, Making a case for multi-agent systems, in: M. Boman, W. Van de Velde (Eds.), Multi-Agent Rationality: Proceedings of the Eighth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-97), Lecture Notes in Artificial Intelligence, Springer, Berlin, 1997.

Michael P. Wellman is an Associate Professor in the Department of Electrical Engineering and Computer Science at the University of Michigan. He received a Ph.D. in Computer Science from the Massachusetts Institute of Technology in 1938 for his work in qualitative probabilistic reasoning and decision-theoretic planning. From 1988 to 1992, Dr. Wellman conducted research in these areas at the USAF's Wright Laboratory. Current research focuses on computational market mechanisms for distributed decision making, and electronic commerce. He is an NSF National Young Investigator, and serves as Executive Editor of the Journal of Artificial Intelligence Research.

Peter Wurman is a Ph.D. candidate in Computer Science and Engineering at the University of Michigan. He has M.S. degrees from the University of Michigan in Computer Science and Mechanical Engineering, and an S.B. degree from the Massachusetts Institute of Technology in Mechanical Engineering. Research interests include multiagent systems, market-oriented programming, and electronic commerce.