

A Trading Agent Competition for the Research Community

Michael P. Wellman and Peter R. Wurman

University of Michigan, Ann Arbor, MI 48109-2110 USA
{wellman,pwurman}@umich.edu
<http://ai.eecs.umich.edu/people/{wellman,pwurman}>

Abstract. We discuss the design of a trading-agent competition, to be held in conjunction with ICMAS-00. This design will be revised based on deliberations of a committee comprising active researchers in the field.¹ This AmEC-99 presentation constitutes an official announcement of the competition. We solicit feedback from the community regarding further operational and design details, and are hopeful that AmEC attendees will consider participating in the competition.

1 Introduction

Trading in electronic markets is increasingly becoming both a commonplace economic activity and a topic of special interest within the AI, Electronic Commerce, and Multiagent Systems (MAS) research communities. Inspired by the success of competitions in various other AI realms,² we propose to organize a “trading agent” competition to be held at the next International Conference on Multiagent Systems (ICMAS-00). Such an event would serve several goals:

1. Spurring research on common problems within the domain.
2. Promoting definitions of benchmarks and standard problem descriptions.
3. Showcasing research in this area to the MAS community and beyond.
4. Having fun.

The purpose of this document is to describe the proposed competition, including its goals, organization, and technical definition. Because there are in fact many ways to structure a competition among trading agents, we present some of the possibilities under consideration, along with underlying design issues that will guide our ultimate choice of competition structures.

¹ Members: Arne Andersson, Dave Cliff, Toru Ishida, Jeffrey Kephart, John Miller, Juan Antonio Rodriguez, Fredrik Ygge, and Edmund Durfee (ex officio).

² The recent RoboCup competitions [12] have been the most prominent. AAAI has held robot competitions at its National Conference since 1992 [1], and the AI Planning Systems Conference held a planning competition at its 1998 meeting (<http://ftp.cs.yale.edu/pub/mcdermott/aipscomp-results.html>). And of course, computer game-playing tournaments have been commonplace for years.

1.1 Market Games

At the core of a trading competition are one or more *market games*, that is, well-defined scenarios whereby agents interact through a market. A market game specification dictates:

1. The set of goods (resource types) to be exchanged through the market.
2. The *market mechanism*, which in turn dictates
 - (a) what the allowable actions are for the various agents, as a function of the market state (which is of course influenced by agent actions),
 - (b) the information made available to the agents by the market, and
 - (c) the outcomes of the market (i.e., allocations, perhaps in terms of a set of trades), as a function of the agents' actions.
3. The value accrued by agents as a function of the outcome allocations.

Market games range from very simple to arbitrarily complex, and on the complex end can vary in their demands for agent competence in strategy, coordination, timing, domain-level reasoning, or other relevant capabilities. Choice of a market game dictates the technical issues that agent designers will face, and thus frames the entire research component of the event. It will undoubtedly be a major factor in the decisions of potential competitors whether to participate, and is thus arguably the most important design decision facing competition organizers. Section 3 presents some particular market games that we may consider for the trading agent competition.

1.2 Competition Overview

The competition would be held at ICMAS-00, in conjunction with a workshop devoted to that purpose. We expect the workshop to last one day, and be held before the conference. We intend to actually run the competition (or at least its final stages) during the workshop. Each game will be supported by an accompanying dynamic display and tools for automated reporting of the results.

We will run the market games on the Michigan Internet AuctionBot platform [21].³ The AuctionBot is a configurable auction server that implements a diverse set of auction rules and thus enables a wide variety of market games. The server has been operational for several years and has matured into a stable auction platform. In addition, the AuctionBot supports participation by software agents through a TCP-based agent programming interface (API).

2 Competition Structure

Many of the contest details remain to be decided. In this section we present some of the issues that are independent of which market games are selected. These issues relate to the role of the workshop in supporting the contest, and the policies of the contest itself.

³ See <http://auction.eecs.umich.edu>. As operators of the competition, our group will not submit trading agent entries.

2.1 Workshop

The purpose of the workshop is to exchange experiences in preparing for the competition, and observations on the results. By holding some of the actual competition at the workshop, we also create a sense of the event itself. However, this would make it a requirement that participants attend the workshop, which might exclude certain “non-traditional” participants, such as high school and college students not working in a research environment. On the other hand, relaxing this restriction and allowing remote participation places further burdens on the contest organizers.

A natural workshop activity would be to elicit presentations by the agent design teams that discuss their architectures and strategies. In the spirit of our research goals, we should encourage disclosure and scrutiny of ideas inspired by the competition process. In fact, we should probably require this as a prerequisite for participation.

Ideally, the contest will result in some artifacts that can be made available to other researchers. The game description and results can be easily compiled in an accessible form. In addition, we could adopt a policy that all information available to the system—including that not disclosed during the trading session itself, such as exact bid streams—be treated as public knowledge after the fact. Finally, we can encourage participants to make available their actual agent source code, as well as other precise behavioral descriptions of their agents.

2.2 Contest

The contest will consist of a small number of market games. We will build a *market-game engine* that is capable of executing and evaluating the games. The number and type of games is yet to be determined, however their executions will follow the same basic pattern. The contest organizers will provide a set of initialization files that detail each agent’s starting conditions for the game (e.g., role, preferences, capabilities, endowments). The engine will initiate the appropriate auctions. The agents will be instantiated with the appropriate initialization files (who initiates the agents is yet to be decided). The engine will monitor the game, and when the game closes will generate a report detailing the performance of each agent.

Within this overall structure, there are a number of open issues.

Contest timing. Ideally, the contest would be run during the workshop itself.

However, even this is open for discussion—we could run the contest before the workshop and use the workshop to discuss the results and agent designs.

Number of game types. More games allow us to explore different issues in the design of market-aware agents. However, too many games can result in thin participation in each. We expect that either one or two games will be selected.

Population size. The structure of the contest can vary with the number of participants. Depending on the number of contestants, we may introduce a

tournament structure, with semi- or even quarter-final rounds before a final. Games will differ on how well they scale with varying population sizes, and we may be associated with preferred population sizes for each game instance.

Game instances. Most of the games will involve multiple instances, randomly generated from some distribution, described in advance to competition participants. We need to design a distribution that is simple to specify, yet sufficiently biased toward interesting game instances.

Number of trials. Should each game be run once, or several times, and with what aggregate performance measure? Because the system is distributed and asynchronous, different runs with the same initial conditions could produce different results. More importantly, several trials may be desirable in order to average over several draws from randomly generated game instances. In games with distinct agent roles, we may want to run the game repeatedly with each agent getting a chance to play each role.

Qualification. It may be worthwhile to hold a qualification round a week or two before the contest in order to avoid any major surprises at the workshop.

There are at least three ways in which qualification could be used:

- The agent is submitted to the organizers at a designated time. No further work is allowed by the designers before the contest.
- The agent is submitted (or executed remotely) at the specified time, but the designers are allowed to continue work. The intent is to allow the design team to make final changes up to the last minute, however we would need some way to ensure that contestants do not submit a working agent that is unrelated to the one they intend to run in the contest.
- The qualification step is provided simply as a chance to check an agent's performance in conjunction with default agents supplied by the organizers. This would provide some confidence to both the organizers and the design team that an agent works.

Visualization. We should have some visualization of the game as it progresses.

Each game will most likely have its own natural visuals.

Prizes. Some researchers will participate simply for fun and (potential) glory. However, real prizes can be motivating, and some sort of presentation ceremony can add to the overall enjoyment. It may be possible to obtain sponsor donations of small prizes.

2.3 Operational Considerations

There are several software components that we have now, or plan to have, to support the development teams:

API with documentation. The API defines the content and format of messages sent to and from the AuctionBot. All messages are encoded as strings, so developing new implementations of the API is not difficult, for those who would prefer to do so.

Agentware. Market-agent foundation classes that encapsulate the API. We currently have beta versions in C++ and Java that will be completed in

time for use by the participants. Agentware will be provided for as many platforms and languages as possible.

Dummy agents. Simple agents provided to give participants some fellow players with which to try out a market game. We might include several types of dummy agents, to provide a sampling of possible contexts. The source code for all dummies will be made available to serve as a foundation on which to build more elaborate agents.

Testing tools. Tools to facilitate setting up a test market game, running it, and generating summary reports. We currently have AuctionBot software components that support these activities, and these tools will be adapted for use by contestants.

There are many practical issues related to running the contest on location. On what machines should the agents run: laptops on location, or remote sites controlled by either the participant or the organizers? If the agents are not controlled by the organizers, the initialization files need to be transferred to agents in a timely manner and the participants need access to a terminal from which to invoke their agent. If participants bring their own laptops, we will need network connections for everyone, and desktops to work from.

Our goal is to maintain a fun and cooperative atmosphere among the participants even while the agents participate in a highly competitive environment. If remote participation is allowed, we may be able to use Internet video conferencing to more directly include the off-site teams.

2.4 Laws and Enforcement

Ideally, game rules will be completely specified and completely enforceable by the market institution. Unfortunately, we cannot rule out the possibility of:

- *Cheating*: rule violations that subvert technological enforcement.
- *System bugs*.
- *Fatal loopholes*: unanticipated behaviors technically consistent with the rules, but threatening to degenerate the competition.

Clearly cheating is intolerable, and if detected would be grounds for disqualification. Bugs would by definition call for ad hoc response. Whether behavior constitutes a loophole is clearly a matter of judgment, and we require some process for exercising such judgments *in situ*.

3 Market Game Scenarios

In the following, we provide capsule summaries of some market games that could be employed in a trading competition. The selection is meant to be suggestive of the range of possibilities, rather than comprehensive. It includes some natural choices, as well as some games from our own research and pedagogy.

3.1 Continuous Double Auction

Perhaps the purest trading scenario conceivable is an abstract market based on the continuous double auction (CDA) mechanism [6]. The CDA is a simple and well-studied auction institution, employed commonly in commodity and financial markets. The “double” in its name refers to the fact that both buyers and sellers submit bids, and it is “continuous” in the sense that the market clears instantaneously on receipt of compatible bids.

The CDA has also been widely employed in experimental economic studies, and even in a prior trading agent competition—conducted at a Santa Fe Institute workshop in 1990 [15]. Over the last ten years, CDA markets have served as a basis for many studies of artificial trading agents (Cliff’s recent example [4] provides an extensive bibliography).

A CDA exercise typically commences with a random assignment of good valuations and endowments to agents, followed by a trading phase. Bidding may take place in a regulated sequence, or asynchronously. Once trading terminates (based on quiescence or a fixed number of bidding iterations), the agents accrue their respective surpluses, and another round—with new assignments of endowments and valuations—begins.

The simplicity and familiarity of the abstract CDA framework presents some distinct advantages as the basis for a trading agent competition. These include ease of explanation and operation, low barriers to entry, consensus understanding of market rules, predictability of behavior, opportunity to build on prior work (on design of both mechanism and agents), and analyzability of outcomes. Given the ubiquity of the CDA institution, there is even a potential to incorporate real-world market data of some sort.

On the other hand, the basic scenario may be too simple and familiar to induce novel and interesting agent designs. Indeed, a lesson from prior experience is that it is difficult to make sophisticated reasoning pay off in this context [8, 15]. For the same reason, the scope of new insights that might be expected from the exercise could be limited.

Some incremental extensions and variants to the basic CDA may overcome some of these disadvantages in part. The possibilities include:

- Complex *activity rules*. The winning trader in the 1990 SFI competition [15] held back until most of the other agents revealed their valuations through bidding behavior, then “stole the deal” at an advantageous price. Activity rules restrict such behavior by requiring agents to participate sufficiently in the price-formation process throughout. Such rules were an important element in the US FCC spectrum auctions [10], and present interesting strategic problems to participating agents.
- Multiple interdependent goods. Valuations that depend on joint holdings of multiple goods present trading agents with a qualitatively distinct strategic problem. We can allocate multiple goods through multiple CDAs, run simultaneously [9] or sequentially [2]. Indeed, most of the problem scenarios discussed below feature this kind of interdependency, but organized with respect to a particular task rather than in the abstract CDA framework.

- Rich information structure. Another way to make the agent problem more interesting would be to provide (perhaps asymmetrically) exogenous sources of information about the other agents’ valuations and endowments. A design challenge would be figuring out how to do this without turning the problem into one essentially of evidential reasoning, rather than trading.

3.2 Scheduling

As mentioned above, one interesting way to enrich a market scenario is to introduce interdependencies in the agents’ values for combinations of goods. We can do this in a purely abstract way by defining nonadditive valuation functions, or in a somewhat more concrete way by defining a task environment that exhibits natural interdependencies among a given set of good types.

For example, if two input resources are both needed to perform a task, then the *complementarity* of these inputs represents a source of interdependency. One very broad and commonplace task environment rife with such complementarities is *scheduling*, where the inputs are units of schedulable resources over fixed periods of time. For instance, consider an agent that needs two hours of factory time to complete a job. If the factory is allocated in one-hour intervals, then the agent has complementary demand for any two adjacent slots.

We have defined a market model for a simple version of scheduling, in which each agent’s job requires a particular number of slots, and achieves variable value depending on how soon the job is completed [20]. For example, in the problem instance depicted in Figure 1, four agents each have one job to perform, with specified deadline and value. Since there is not enough factory time to complete all the jobs, the agents must compete through the market for the available slots. One natural way to run the market is through a simultaneous ascending auction mechanism [11], in which agents bid for individual slots of interest, attempting to acquire sufficient factory time through combinations of bids.

An advantage of scheduling as a competition domain is that it is a common problem—many tasks involve some element of scheduling, or resources with some temporal aspect. The model proposed here is relatively simple, yet includes interdependencies rendering the agents’ strategic problem decidedly nontrivial. Disadvantages include the fact that it can be quite difficult to obtain efficient allocations; indeed, we have found it quite easy to construct examples that lead to poor outcomes (i.e., slots are not allocated to the most valuable jobs) for seemingly reasonable agent strategies.

3.3 Supply Chain Formation

Another interesting source of interdependency is *precedence* in resource production. A scenario that captures precedence in a fairly pure form is that of *supply chain formation*, where agents contract to perform tasks for each other, with tasks related through hierarchical dependencies. We represent these relationships in *task dependency networks* [18], as shown in Figure 2.



Fig. 1. A factory scheduling market. Lines connecting the agents to time slots represent one feasible allocation. An AuctionBot-based demonstration of the example, in which the subject plays agent 1, is accessible at <http://auction.eecs.umich.edu/demos/factory.html>.

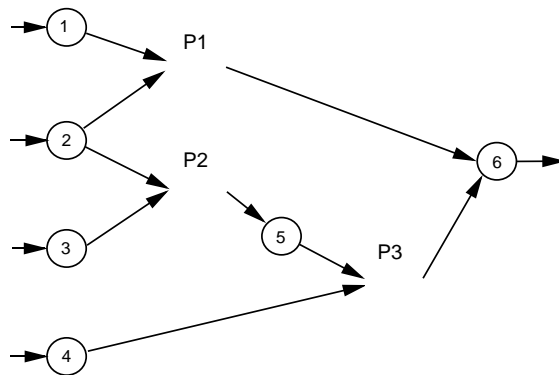


Fig. 2. A simple task dependency network. There are two ways to accomplish the final task (6): via producer P1, or the chain of P2 and P3.

In the network, rectangles represent agents, and circles the tasks (or other kinds of good) they can provide or acquire. The interesting class of agents are *producers* (labeled P1, P2, P3 in the figure), which can provide their output good only if they acquire their inputs. They must simultaneously negotiate to buy their inputs and sell their outputs, even though doing so exposes them to some risk of infeasibility or wasted inputs.

Although we can define simple bidding policies that when applied by all agents tend to produce good results [19], the problem in general is quite difficult. In fact, we have employed this scenario in classroom exercises, and in 10 trials of a 25-agent network, the class failed to form a complete supply chain through decentralized bidding, even though the network was dense in solutions. In a much larger number of trials with student-written trading agents, no instance with a heterogeneous set of agents was successful. A few of the agents (approximately 5 out of 80) were able to form supply chains when their policies were replicated throughout the network.

Although supply chain formation is a significant problem for MAS research and electronic commerce, the pure task-dependency network model may be too brittle to serve as the basis for a trading agent competition. Nevertheless, it may be desirable to incorporate some aspects of the model in whatever scenario is adopted for the competition.⁴

3.4 Widget Economy

The widget economy combines some of the features of the scheduling and supply-chain games, in a two-stage production model. Because the game involves multiple interdependent resources and “closes the loop” between production and consumption, it captures some of the flavor of general-equilibrium models.

The widget economy was originally designed as part of a class exercise, and has been used in several courses.⁵ There are three distinct agent roles, depicted schematically in Figure 3. Agents are characterized by the goods they provide and acquire.

- *Consumers* are endowed with raw materials and derive utility by consuming widgets.
- *Component Manufacturers* purchase raw materials and convert them into components.
- *Widget Assemblers* purchase components and combine them to produce widgets, which they then sell to the consumers.

There are three types of widgets in the world, each of which is assembled from a different combination of the four components. A component manufacturer’s production technology is specified by the amount of raw materials that

⁴ Descriptions of the exercises are available at <http://ai.eecs.umich.edu/people/wellman/courses/eecs498/w97/PS3.html> and <http://ai.eecs.umich.edu/people/wellman/courses/eecs492/w98/PS6.html>.

⁵ <http://ai.eecs.umich.edu/people/wellman/courses/eecs498/w97/PS2.html>

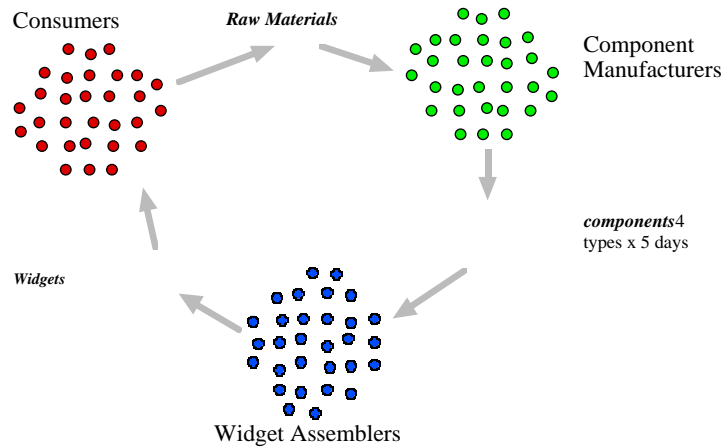


Fig. 3. Flow of resources among agent types in the widget economy.

are needed to produce each component type. A widget assembler's technology is described by the number of components that go into each widget. In addition, each producer has capacity constraints limiting daily production.

The overall economy essentially computes—in a decentralized manner—a five day production plan that converts raw materials into widgets. The consumers attempt to maximize their utility. The component manufacturers and widget assemblers, on the other hand, strive to maximize profit.

However, both types of producers have heavy machinery that makes it impractical to switch products during the business day—they are allowed to make only one type of component or widget on each of the five days. Moreover, a widget assembler cannot use a component until the day after it is produced.

The basic plot is simple. Markets are opened to facilitate the formation of a production plan. After all of the markets close, the production plan is executed. Raw materials are available as contracted on Day 0 of the execution, and the widgets get delivered at the end of Day 5. The consumers may then eat the widgets at their leisure.

The goods are traded in double auctions with continuous or periodic clears. Agents are allowed to buy or sell any of the goods while the markets are open. All of the markets close at the same prespecified time, after which the transactions are settled. *Infeasibility*, that is, the inability of an agent to settle its contracts from either its endowments, purchases, or production technology, is penalized heavily.

The widget economy is attractive for a number of reasons, not the least of which is that we have experience running it. The game scales well with the number of participants and is relatively resistant to coalition formation and deadbeat agents. The game has several roles, each of which faces nontrivial strategic problems. Finally, when utility functions, endowments, and production technologies

are uniform, the general equilibrium analysis is tractable and provides a benchmark by which to evaluate group performance.

Perhaps the biggest drawback of this game is that the story line is complex. Perhaps our success in explaining it to undergraduates bodes well for our ability to describe it to the research community.

For the competition, the game can be adjusted in a variety of ways. We can increase or decrease the number of components. We can vary the scheduling horizon, or move from a fixed horizon to a rolling horizon game. We can vary the endowments, utility functions, or production technologies assigned to the agents.

3.5 Shopping

Comparison shopping facilities, some based on AI/agent research [5], are becoming commonplace on the world-wide web. Automated negotiation between buyer and seller seems a next obvious step. Although two-player bargaining about a single item is probably not a rich enough problem to sustain interest for a trading competition, more complex shopping scenarios involving combinations of goods, multiple options, and dynamic presentation of opportunities may well make for an interesting scenario.

We have sketched a version of this more complicated shopping scenario, called “The Big Getaway”. In this game, shoppers (i.e., the trading agents) want to quickly assemble a weekend getaway for two—consisting, for example, of plane flights, a hotel room, theater tickets, etc. Agents would face constraints (e.g., the hotel must be in the same city as the destination airport), as well as preferences (for different cities, hotel qualities, transportation arrangements, and entertainment options). These preferences may be multi-faceted (class, mode, duration, and timing of transportation), include interdependencies (desirability of renting a car depends on hotel location), and of course may vary across agents (some appreciate the symphony more than others).

The goods may become available over time according to some specified distribution, allocated through auctions, perhaps with different auction mechanisms for different types of goods. For example, we might have airline tickets offered according to some mysterious dynamic reserve-price policy, sell hotel rooms through one-sided auctions, and trade entertainment tickets through two-sided auctions (perhaps distributing tickets randomly to the agents at the start). Policy questions include whether and how to allow reselling, and what information to provide for agents as a basis for their predictions of future market conditions.

3.6 Others

The set of market game scenarios enumerated above is far from exhaustive. The literature includes scenarios developed by researchers for a range of purposes—including auction tournaments (e.g., Fishmarket [13], an electronic Dutch auction modeling daily sales in a fishing port), experimental economics (e.g., a simple

rail scheduling scenario developed to test combinatorial mechanisms [3]), task allocation in multiagent systems (e.g., TRACONET [16], a system for distributed vehicle routing), or abstract models of multiagent planning problems (e.g., the task- and state-oriented domains of Rosenschein and Zlotkin [14]). Any of these (or many others) could provide a suitable basis for a trading agent competition. We would be interested in suggestions for how to incorporate interesting features of other scenarios in the market games actually chosen for the competition.

The focus on interdependencies in most of the market games presented above may suggest to some that some kind of combinatorial auction [17] or other multidimensional mechanism is warranted. By allowing agents to bid directly on combinations of goods or features, we can perhaps obviate much of the strategic complexity inherent in trading across separate but independent market mechanisms. However, whereas we agree that expressly multidimensional mechanisms offer great promise for automated trading, we expect that their scope must ultimately be limited, and that there will remain a significant role for traders capable of dealing with separate mechanisms for interdependent goods. It is certainly the case that such behavior is necessary in the current state of practice.

4 Design Issues

The potential market games should be evaluated by how well they promote the contest goals. We wish to choose a set of games which, individually or jointly, satisfy the following criteria:

1. Presents complex and interesting strategic issues.
2. Encourages general strategies.
3. Is easy to explain and understand.
4. Scales with the number of participants.
5. Reflects markets that are likely to appear in the real-world.

For each game that is chosen, the following decisions need to be made.

Roles. Some games put agents in different roles. Should a contestant be required to submit agents for each role? If not, do we use our dummy agents to fill in the market when certain roles have thin participation?

Number of submissions. Should a person be allowed to submit more than one agent? If so, should they participate in the same game instances? Can an agent be designed to do poorly so that the contestant's other agent wins?

Number of agents. How many agents will be run in a given round? If there are not enough contestants, will more than one copy of an agent be used? Or will dummy agents be used to fill in the holes?

Agent groupings. Should the entered agents play games with each other, or with provided dummy agents? The former is certainly more competitive (and perhaps more realistic), and does not impose extra work on the organizers.

Communication. What communication is allowed between agents? If someone has agents in several roles, should they be allowed to attempt to coordinate their behavior? Can a no-communication rule be enforced?

Tournament. An agent’s performance clearly depends on which agents it plays against. Do we go with luck of the draw, or impose a round-robin structure to equalize this factor? We could even consider tournaments organized in an evolutionary regime.

Scoring. How will the game be scored? Will the scores from earlier rounds carry over in any way to the finals? If a game involves multiple roles, will there be one winner in each role, one winner overall, or both (like gymnastics)?

Scoring—or more generally, incentive engineering—is a particularly tricky issue. Although as part of the market game we define a value function for the agent, this does not necessarily coincide with the objectives from the perspective of the *agent designer*. For example, in a winner-take-all contest, the objective is to maximize the probability of getting the highest score, which can lead to strategies quite different from that maximizing the expected score. We should find ways to make clear that although this is called a “competition”, the actual game is not zero-sum, and to design the scoring rules so that highly risky strategies are extremely unlikely to succeed.

5 Conclusion

We believe that the time is ripe for a trading agent competition for the research community. With careful design, and the support and participation of active researchers in the field, we are hopeful that it can prove a worthwhile endeavor. We welcome feedback from the community about the organization of this effort, and encourage all readers to start contemplating their agent designs.

Acknowledgments

We have benefited enormously from brainstorming of colleagues at the University of Michigan. Fredrik Ygge provided pivotal suggestions regarding the organization of this effort. Robert Axelrod, Klaus Fischer, and the competition organizing committee have also offered useful guidance, most of which has not yet been reflected in this document.

References

1. Pete Bonasso and Thomas Dean. A retrospective of the AAAI robot competitions. *AI Magazine*, 18(1):11–23, 1997.
2. Craig Boutilier, Moisés Goldszmidt, and Bikash Sabata. Sequential auctions for the allocation of resources with complementarities. In *Sixteenth International Joint Conference on Artificial Intelligence*, Stockholm, 1999.
3. Paul J. Brewer. Decentralized computation procurement and computational robustness in a smart market. *Economic Theory*, 13:41–92, 1999.
4. Dave Cliff. Evolving parameter sets for adaptive trading agents in continuous double-auction markets. In *Agents-98 Workshop on Artificial Societies and Computational Markets*, pages 38–47, Minneapolis, MN, May 1998.

5. Robert Doorenbos, Oren Etzioni, and Daniel S. Weld. A scalable comparison-shopping agent for the world-wide web. In *First International Conference on Autonomous Agents*, pages 39–48, 1997.
6. Daniel Friedman. The double auction market institution: A survey. In Friedman and Rust [7], pages 3–25.
7. Daniel Friedman and John Rust, editors. *The Double Auction Market*. Addison-Wesley, 1993.
8. Dhananjay K. Gode and Shyam Sunder. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101:119–137, 1993.
9. Junling Hu and Michael P. Wellman. Online learning about other agents in a dynamic multiagent system. In *Second International Conference on Autonomous Agents*, pages 239–246, Minneapolis, 1998.
10. R. Preston McAfee and John McMillan. Analyzing the airwaves auction. *Journal of Economic Perspectives*, 10(1):159–175, 1996.
11. Paul Milgrom. Putting auction theory to work: The simultaneous ascending auction. Technical Report 98-0002, Department of Economics, Stanford University, December 1997.
12. Itsuki Noda, Shóji Suzuki, Hitoshi Matsubara, Minoru Asada, and Hiroaki Kitano. RoboCup-97: The First Robot World Cup Soccer Games and Conferences. *AI Magazine*, 19(3):49–59, 1998.
13. Juan A. Rodríguez-Aguilar, Francisco J. Martín, Pablo Noriega, Pere Garcia, and Carles Sierra. Competitive scenarios for heterogeneous trading agents. In *Second International Conference on Autonomous Agents*, pages 293–300, Minneapolis, 1998.
14. Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press, 1994.
15. John Rust, John Miller, and Richard Palmer. Behavior of trading automata in a computerized double auction market. In Friedman and Rust [7], pages 155–198.
16. Tuomas Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the National Conference on Artificial Intelligence*, pages 256–262, Washington, DC, 1993. AAAI.
17. Tuomas Sandholm. Approaches to winner determination in combinatorial auctions. *Decision Support Systems*, to appear.
18. William E. Walsh and Michael P. Wellman. A market protocol for decentralized task allocation. In *Third International Conference on Multiagent Systems*, pages 325–332, Paris, 1998.
19. William E. Walsh and Michael P. Wellman. Efficiency and equilibrium in task allocation economies with hierarchical dependencies. In *Sixteenth International Joint Conference on Artificial Intelligence*, Stockholm, 1999.
20. William E. Walsh, Michael P. Wellman, Peter R. Wurman, and Jeffrey K. MacKie-Mason. Some economics of market-based distributed scheduling. In *Eighteenth International Conference on Distributed Computing Systems*, pages 612–621, Amsterdam, 1998.
21. Peter R. Wurman, Michael P. Wellman, and William E. Walsh. The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In *Second International Conference on Autonomous Agents*, pages 301–308, Minneapolis, 1998.