A testbed for multi-agent contracting for supply-chain formation

John Collins
Dept of Computer Science and Engineering
University of Minnesota
jcollins@cs.umn.edu

Maria Gini
Dept of Computer Science and Engineering
University of Minnesota
gini@cs.umn.edu

ABSTRACT

We are interested in the problem of multi-agent contracting, in which customer agents must solicit the resources and capabilities of other, self-interested agents in order to accomplish their goals. Goals may involve the execution of multi-step tasks, in which different tasks are contracted out to different suppliers. We have developed a testbed that allows us to study decision behaviors of agents in this context. It can generate sets of plans with known statistical attributes, formulate and submit requests for quotations, generate bids with well-defined statistics, and evaluate those bids according to a number of criteria. Each of these processes is supported by an abstract interface and a series of pluggable modules with many configuration parameters. Data collection and analysis tools round out the package.

1. INTRODUCTION

The business-to-business (B2B) e-commerce market is expected to expand rapidly, with the global market expected to exceed \$7.29 trillion in 2004, according to Gartner Group research. A recent separate study from Boston Consulting Group predicts productivity gains from B2B e-commerce will equal $1\% - 2\%$ of sales by 2004 and 6% by 2010.

Firms can cut costs and improve efficiency by moving online. Instead of fulfilling orders from warehouses, companies will look for manufacturers that can build on demand in order to meet consumers demand for make-to-order products. More production processes will be outsourced, making supply chains longer and more convoluted. The increased complexity will be compounded by accelerated production schedules which demand tight integration of all processes. Thus, the field is ripe for the introduction of systems that automate logistic planning among multiple entities such as manufacturers, part suppliers, shippers, and specialized subcontractors.

Current e-commerce systems typically rely on either fixed-price catalogs or auctions. Companies usually work with prequalified suppliers and buyer-supplier relationships depend on factors such as quality, delivery performance, and flexibility as opposed to just cost [10]. In addition, current e-commerce systems do not have any notion of time (except for domain specific systems such as SABRE used in the travel industry). Time plays a fundamental role in supply-chain formation and management, since many products are made up of different parts and require multiple suppliers who have to coordinate their work.

We are interested in understanding how a community of heterogeneous, self-interested agents, can make commitments and carry out plans that require multiple tasks and coordination among multiple agents. We have proposed a market architecture [5] and we have implemented prototypes of both the market architecture and the agents. We call this system MAGNET (Multi AGent NEgotiation Testbed). MAGNET provides support for a variety of types of transactions, including complex multi-agent contract negotiations with temporal and precedence constraints.

This paper is organized as follows: Section 2 describes the environment of MAGNET agents, and the basic activities and roles of agents in that environment. Section 3 describes our experimental implementation of a customer agent that we are using to explore agent decision processes. Section 4 describes the implementation of abstract supplier agents. Section 5 gives some examples of the types of studies supported by this framework. Section 6 describes related work, and Section 7 concludes and outlines our future plans and open problems.

2. THE MAGNET SYSTEM

MAGNET gives an agent the ability to use market mechanisms (auctions, catalogs, timetables, etc.) to discover and commit resources needed to achieve its goals. We assume that agents are heterogeneous and self-interested, and may be acting on behalf of different individuals or

To appear in Workshop on Agent-Based Approaches to B2B, May 2001.

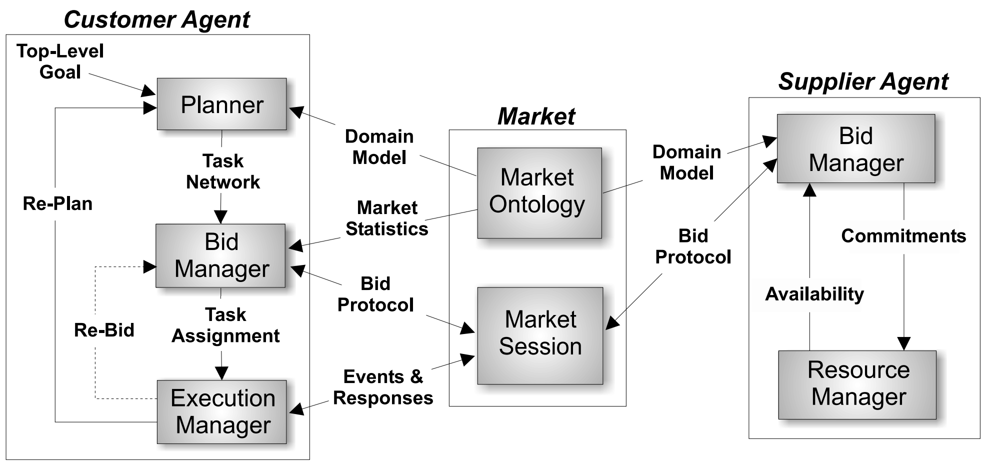

Figure 1: The MAGNET architecture

commercial entities who have different goals and different notions of utility. Although we use auction mechanisms, the problem MAGNET must solve is a combination of a scheduling problem and a combinatorial auction problem.

Agents may fulfill one or both of two roles with respect to the MAGNET architecture, as shown in Figure 1. Customer agents pursue their goals by formulating and presenting Requests for Quotations (RFQs) to Supplier agents through the market infrastructure [5]. Customers have goals that they themselves cannot satisfy, either because they lack the abilities, or the resources to carry out at least some of the operations. Suppliers have resources to offer, and are willing to make those resources available in a way that maximizes their value.

The RFQ specifies a task network that includes task descriptions, a precedence network, and possibly other time constraints. Customer agents attempt to satisfy their goals for the least net cost, where cost factors can include not only bid prices, but also goal completion time and risk factors. More precisely, these agents are attempting to maximize the utility function of some user, as discussed in detail in [3].

Supplier agents attempt to maximize the value of their resources by submitting bids in response to those RFQs. Bids specify what tasks they are able to undertake, when they are available to perform those tasks, and at what price. Bids may specify combinations of tasks with a single price, and may also include prices on individual tasks. Prices for multiple tasks can include a discount or a premium.

As an example, let's imagine we need to do a site preparation for installing a large-scale server. Figure 2 shows a plan to complete the site preparation. Our plan is complicated by a couple of factors. The server must be ordered ahead, and will arrive one week after we order it. Because it is the most expensive part of the installation, we do not want to leave it idle, and so it must be installed immediately when it arrives. Also, there is a network boom in our area, and people to do wiring are hard to find. We may have to wait.

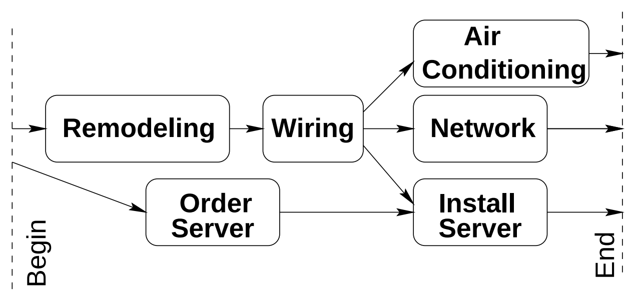

Figure 2: Plan for preparing a site for installation of a large-scale server.

3. A CUSTOMER AGENT

We now focus on the structure and responsibilities of a Customer agent in the MAGNET environment. As indicated in Figure 1, the basic operations are planning, bidding, and plan execution. We have implemented a simple Planner that generates random plans with well-defined statistics, and a Bid Manager with a fairly rich implementation of tools for composing RFQs and selecting bids. The Execution Manager is not yet implemented.

3.1 Planner

The Planner's task is to turn high-level goals into executable plans, represented as task networks. A task network consists of a set of task descriptions, the temporal constraints among them, and possibly nonzero delays between tasks, to cover communication and transportation delays.

The planner in the testbed generates tasks by selecting randomly from a library of task types, and then creates random precedence relations among them. It can also accept pre-defined plans. We expect that in many domains, plans will be chosen from a library or defined by a human user rather than being generated by a general-purpose planner.

The definitions of tasks must be shared among the agents. That is why we show the communication of the Domain Model from the Market to the Agents in Figure 1. This model includes not only the task definitions, but statistics (presumably collected by the market) about each task type. These statistics include expected duration and variability, expected price and variability, and resource availability data.

In our site-preparation example, we would find out from the market that wiring resources are thin. This data is then included in the plan received by the Bid Manager.

The plan generated by the Planner is a central data structure throughout a MAGNET system. The Bid Manager uses it to generate RFQs and to evaluate and record resource commitments and timing data, and the Execution Manager uses it to monitor and repair the ongoing execution of the plan. Part or all of the plan is included in a RFQ. In fact, each of the other components can be characterized by how it uses, decorates, extends, or updates the plan.

3.2 Bid Manager

The Bid Manager is responsible for ensuring that resources are assigned to each of the tasks of a plan, that the assignments taken together form a feasible schedule, and that the cost and risk of executing the plan is minimized. This cost must also be less than the value of the goal at the time the goal is reached.

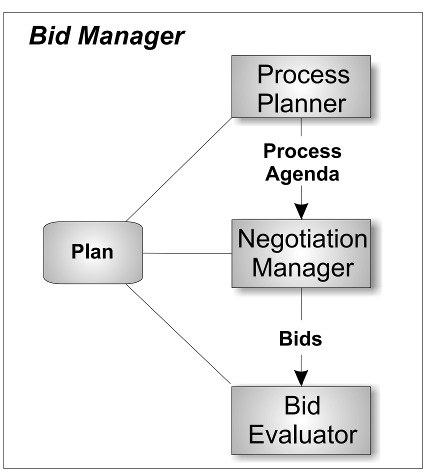

Figure 3: The Bid Manager

When the Bid Manager is invoked, some tasks in the plan may already be assigned. This can occur because the Execution Manager may use the Bid Manager to repair a partially-completed plan in which previously determined assignments have failed, because the agent will perform some of the tasks itself, or because bidding is being carried out in multiple stages. For example, the company may be able to set up the network, and so it might not contract for that task.

The Bid Manager must construct and issue a RFQ, evaluate bids, and accept bids in order to carry out its responsibilities. The high-level structure of the Bid Manager is shown in Figure 3.

3.2.1 *Process Planner*

The Process Planner creates the high-level agenda for the Bid Manager. A primary responsibility is to allocate time to negotiation and plan execution. The current version is really just a placeholder that reads an agenda from a configuration file or a user interface. In the future it will be responsible for deciding which markets to use, when to consult local catalog and timetable databases, and how to break up the plan accordingly. If the plan has alternative branches, it may also decide which alternatives to pursue and in what order. For example, it may decide to solicit bids on a high-value but risky approach, and if that fails to fall back on a lower-value but safer alternative. It could also decide to defer taking bids on later tasks until earlier tasks were underway or even completed. This is standard practice in many industries. In our site-preparation example, we might decide to wait until the wiring was underway before ordering the server.

3.2.2 *Negotiation Manager*

The Negotiation Manager handles the actual bidding process. Its overall job is to decorate the plan with a feasible, minimum-cost set of resource assignments. It uses the Bid Evaluator to decide among alternative bid combinations.

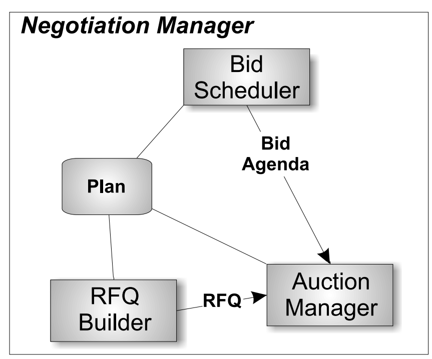

Figure 4: The Negotiation Manager

The Negotiation Manager is further broken down into a set of components, as shown in Figure 4. The Bid Scheduler assembles a schedule for the bidding process, possibly subdividing the time allocated by the Process Planner, and adds items to the agenda to drive the Auction Manager. Dividing the bidding process into multiple phases can be an important strategy to reduce the level of uncertainty in the plan. For example, we might not want to take bids on the air conditioning for our site-preparation task until we have firm dates for the wiring. We'll discuss an example of multi-phase bidding in Section 5.

Several different versions of the Bid Scheduler have been implemented to experiment with different strategies. Ultimately it will be up to the Process Planner to decide which strategy (or strategies) to use, and configure the Bid Scheduler accordingly through its agenda entries.

Before bids can be solicited in a market, an RFQ must be composed. The RFQ is a structure that contains some portion of the plan data (tasks and precedence relations) as determined by the Bid Scheduler, along with a set of scheduling constraints. The primary role of the RFQ Builder is to determine those scheduling constraints. Information comes from several sources:

- From the Planner, we have a set of tasks and their precedence constraints. This information is contained in the plan.
- From the Market, we have statistical information about duration and variability for the different task types. We also have information about resource availability and the number of vendors who are likely to bid on tasks of this type.
- From the Process Planner, we have the overall schedule for the execution of the plan.
- From the Bid Scheduler, we know which tasks are to be advertised for bid in the current RFQ.

The primary goal of the RFQ Builder is to produce an RFQ that will solicit the most advantageous set of bids possible. The approach we take is to find a balance between giving maximum flexibility to suppliers, ensuring that the resulting bids will combine feasibly, and ensuring that the job will be completed by the deadline. We do this by setting early-start and late-finish times in the RFQ for each task.

Figure 5 shows two alternative ways to schedule and compose the RFQs for our site preparation project. In version A, we believe we have 5 weeks to finish our site, and the only scarce resource is wiring. Therefore, we allow 3 weeks for the one-week wiring job, and we are guaranteed that if we receive bids on all tasks, they can be combined feasibly. In version B, we are interested in finishing the site as soon as possible. Therefore, we bid out the remodeling and wiring first in RFQ B1, and we bid out the remainder of the tasks in RFQ B2 after we get a bid that finishes the wiring by the end of week 3.

The Auction Manager interacts with the Market and/or other agents to solicit bids. Different versions of the Auction Manager can be implemented to interact with different market environments. We have a version that uses a MAGNET market to solicit bids, and one that uses a set of in-process simulated Supplier agents directly to generate bids for testing purposes. The latter version is useful for doing large statistical studies where throughput is a critical factor.

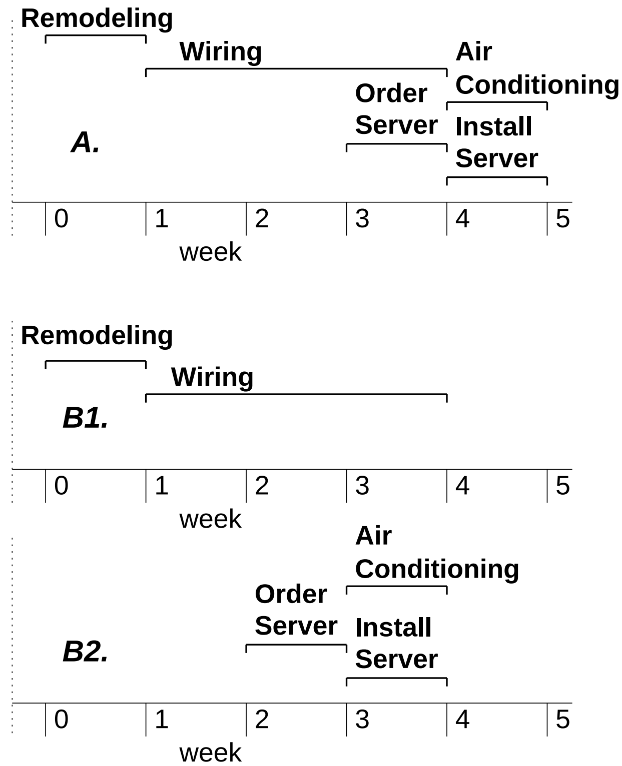

Figure 5: RFQ Example

3.2.3 *Bid Evaluator*

A Bid Evaluator is a search engine that takes a plan and a set of bids, and attempts to find an optimal or near-optimal mapping of bids to tasks, respecting temporal constraints. It must do this within the period of time allocated by the Process Planner, which may have been subdivided by the Bid Scheduler.

We have implemented two evaluators. One is based on Integer Programming, and the other is a highly modular Simulated Annealing (SA) search engine [4].

The Integer Programming (IP) solver operates in two phases. The first phase generates basic bid-compatibility constraints, and then walks all paths of length 2 or greater in the precedence network, across all compatible bid combinations, to discover feasibility constraints. These are then packaged up and sent off to an external IP solver.

The core part of the simulated-annealing engine is similar to the one described in [15]. Starting with a plan and a set of bids, we generate and evaluate bid mappings until one of several stopping conditions holds. These include failure to find improvement for a configurable number of iterations, expiration of the deliberation time limit, and lack of mappings that have any untried expansions. We have described this in detail, along with experimental performance data, in [4].

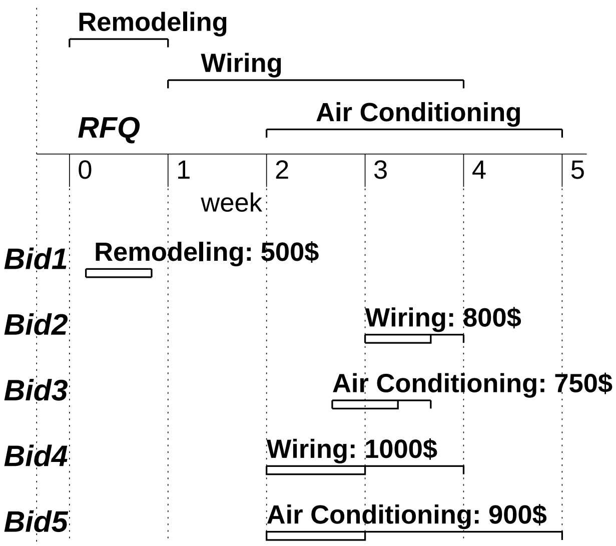

Figure 6: Bid Example

Figure 6 shows a very small example of the problem the Bid Evaluator must solve. We composed the RFQ with a large overlap between the wiring and air conditioning tasks, perhaps because we believed there would be large numbers of bidders with a wide variation in lead times. Bid 2 indicates wiring could start at the beginning of week 3, would take 3 days, and the supplier was willing to shift that out 2 more days to accommodate our schedule. Bid 3 shows that air conditioning could start partway through week 2, would take 3 days, and needed to finish partway through week 3. Clearly these two bids cannot be combined. Bid 4 shows a more expensive wiring person who could start earlier, but needs a week to finish. This can be combined with Bid 3, but with no slack to accommodate contingencies. Bid 5 gives us a large enough time window for the air conditioning task to be combined with either Bid 2 or Bid 4. The best combination appears to be Bid1, Bid2, Bid5.

4. SUPPLIER AGENTS

Since our primary interest has been in the workings of the Customer agent, our Supplier agents are currently fairly simple-minded entities. They receive RFQs, and they respond by submitting bids. They do not maintain resource schedules, and they have no persistent identity. The basic structure is shown in Figure 7. Each of these three layers is implemented as an abstraction with multiple implementations.

A Bid-Set Generator generates sets of bids and returns them to the Customer agent. Example Bid-Set Generators include one that always bids on certain task types if they are present in the RFQ, one that generates a random set of bids, and one that extends the random set generator by attempting to generate a set that covers all tasks in the RFQ.

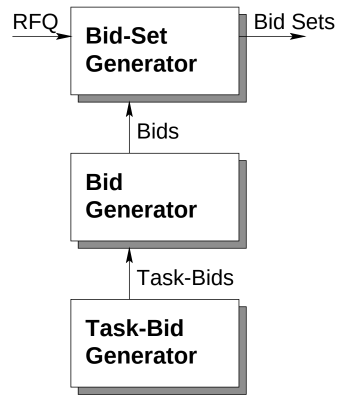

Figure 7: Simple Supplier Simulation

A Bid Generator generates a single bid, possibly containing multiple individual task-bids. The average sizes, and the degree of size variability, of the bids produced are determined by configuration parameters, and in some cases by the structure of the plan and the type of Bid Generator selected. We have implemented Bid generators that can generate bids for certain types of tasks, random collections of tasks, or sets of tasks that are connected by precedence relations. An obvious extension would be to generate role-based bids in the sense of [11].

A Task-Bid Generator produces a bid for a single task. The bid specifies the task to be performed, the expected duration of the task, and early start and late finish time window data. In most cases it must also assign a cost to the task, which the Bid Generator will use in composing the overall cost for the bid. The duration and cost are selected from random distributions specified in the task-type description. The early-start and late-finish times are also randomly generated from the resource-availability data in the task-type description. The constraints on the time window for the Task-Bid come from two sources: (1) the time window specified in the RFQ, and (2) the times already specified in other Task-Bids for tasks that are immediate predecessors or successors of the current task. If the Task-Bid generator cannot fit the requested task into the time window, it fails to produce a result, and the bid will not include that particular task.

5. THE MAGNET TESTBED

Experimental research in this area requires a simulation environment that is sufficiently rich to be easily adapted to a variety of experimental purposes, while being sufficiently straightforward to support clear conclusions. MAGNET is not a complete simulation of a working market environment. Instead, it is focused on the process of determining the form and content of Requests for Quotations (RFQs), on the management of the bidding process, and on the evaluation of bids submitted by potential suppliers. It has the ability to generate plans with well-defined statistics, or to accept hand-built plans or plans extracted from real-world data. Bids are generated by a community of abstract suppliers, again with well-defined statistics. All the major decision processes are driven by plug-in components, with documented APIs and a great wealth of configuration parameters. Data collection capabilities are well-suited to statistical studies.

5.1 Design Principles

In order to maximize the usefulness of the MAGNET testbed as a research tool, we have adopted several design principles that make it easy to plug together and reconfigure, and that enhance its transparency. Examples are:

1. The system is written in Java, and has been tested on multiple platforms. This makes it easy to use on whatever you happen to be sitting in front of.
2. All the major behavioral modules are written as abstract classes, with (at least potentially) multiple implementations that can be "plugged in" to implement a particular behavioral variant.
3. Virtually every feature of the system is selectable and configurable from a configuration file, and many of them can be viewed and changed from a user interface. This includes the choice of behavioral plug-ins.
4. The interface between the agents and the Market is also abstracted. This allows connection with multiple types of markets (such as one that looks up price and availability info from a catalog or timetable) and through multiple communications protocols.
5. Much of the activity of the agent is agenda-driven, and development and maintenance of the agenda is an important activity in its own right. Agenda items can select plug-ins, update configuration details, evaluate options, interact with the market or other agents, update the agenda, and record results.
6. A pervasive logging and data collection system allows for both detailed examination of behavior and the generation of experimental data. The level of logging is a configuration parameter, and the various logging levels have well-defined meanings.

The system in its current form is useful for several types of studies. Recent work includes experiments with bid evaluation performance, and studies of the RFQ composition problem. Our longer-term goal is to support studies of mixed-initiative decision making with experienced human users in realistic market simulations.

5.2 Bid Evaluation

To study bid evaluation, we are able to control a wide range of conditions, including:

- Composition of the generated plans: number of tasks, task types (which in turn controls duration variability and probability of bids), and the density of the precedence network,
- Structure of the RFQ: Whether it covers the whole plan, amount of slack in the schedule, and the degree to which bids are allowed to violate precedence relations,
- Number and size of bids, composition of bids: random selections, contiguous task sets, role-based task sets,
- Type of search used, search parameters,
- Bid selectors and evaluators, evaluation parameters.

The testbed supports a number of measurements for evaluating search performance, including search effort, anytime performance, and solution quality, along with counts of solved, unsolved, and known unsolvable problems encountered. Output is in a form that can be used by a standard spreadsheet, or Matlab in the case of anytime performance data.

An important ongoing effort along these lines is learning how to make bid evaluation work effectively in a mixed-initiative environment. We have studied the implications of Expected Utility Theory in the MAGNET environment [3]. We are currently developing and evaluating evaluators to assess risk, and user interface strategies to support collaborative evaluation and decision-making between a MAGNET agent and its user.

5.3 RFQ Composition

The RFQ composition problem appears to be highly dependent on the characteristics of the market. For that reason, we are closely studying one particular market, international shipping, in hopes of developing a set of data that can support realistic simulation. These data include numbers of likely bidders, likelihood of bidding, specialized vs. full-service suppliers, lead times, and correspondence between bids and actual performance. Further complications arise from standard practices such as capacity consolidation, subcontracting, and the variety of contract terms that are used in a typical supply chain. We are working with North Star Import-Export, a local freight forwarding company, to develop our understanding in this area.

Preliminary results indicate that, given some reasonable number of bidders, some amount of overlap in the task time windows between successive tasks gives better results than a RFQ specification that guarantees that all bids will combine feasibly. We have implemented several different plug-in versions of the RFQ Builder in order to test alternative approaches.

Our goal is to develop a sufficiently realistic simulation of an actual market to support evaluation of MAGNET agent performance by personnel who are experienced in that market. In the shipping domain, some market data can be taken from published timetables, and we will plug in bidders that operate directly from these timetables. There are also Web-based resources such as `www.freightwise.com` that could support supplier-agent wrappers, and which are a good source of availability and pricing data.

6. RELATED WORK

Markets play an essential role in the economy, and market-based architectures are a popular choice for multiple agents (see, for instance, [2, 19, 23, 20]. Most market architectures limit the interactions of agents to manual negotiations, direct agent-to-agent negotiation [18, 6], or various types of auctions [24].

Existing architectures for multi-agent virtual markets typically rely on the agents themselves to manage the details of the interaction between them, rather than providing explicit facilities and infrastructure for managing multiple negotiation protocols. In our work, agents interact with each other through a market. The market infrastructure provides a common vocabulary, collects statistical information that helps agents estimate costs, schedules, and risks, and acts as a trusted intermediary during the negotiation process.

Auctions are becoming the predominant mechanism for agent-mediated electronic commerce [9]. AuctionBot [24] and eMEDIATOR [17] are well known examples of multi-agent auction systems.

The determination of winners of combinatorial auctions [14] is hard. Methods for improving the efficiency of combinatorial auctions have been developed in the last few years, among others, by Sandholm [17] and Fujishima [7]. Mixed integer programming has been demonstrated to work extremely well even on large problems by Andersson [1].

Most work in supply-chain management is limited to strict hierarchical modeling of the decision making process, which is inadequate for distributed supply-chains, since each organization is self-interested, not cooperative. Walsh et al [21] study combinatorial auctions for problems in supply chain, but ignore time constraints. When they study decentralized scheduling [22] they limit their study to the scheduling of a single resource. MAGNET agents have to deal with multiple resources, the bidding process is used as a way of obtaining the use of resources an agent does not have. Customer agents have also to ensure the scheduling feasibility of the bids they accept, and must evaluate risk as well.

Agents in MASCOT [16] coordinate scheduling with the user, but there is no explicit notion of payments or contracts, and the criteria for accepting/rejecting a bid are not explicitly stated. Their major objective is to show the advantage of using lateral coordination policies that focus on optimizing schedules locally through exchange of temporal constraints. Our objective is to negotiate contracts with suppliers that optimize customer's utility.

Andersson [1] proposes integer programming for winner determination in combinatorial auctions. The major difference is that in the cases studied for combinatorial auctions, bid allocation is determined solely by cost. Our setting is more general. Our agents have to cover all the tasks, ensure feasibility of the bids they accept, and reduce scheduling risk.

Because the search space for combination bids with temporal constraints is huge, we have chosen to explore a simulated annealing framework. Since the introduction of iterative sampling [12], a strategy that randomly explores different paths in a search tree, there have been numerous attempts to improve search performance by using randomization. Randomization has been shown to be useful in reducing the unpredictability in the running time of complete search algorithms [8]. Our experimental results [4] show that our bid selection algorithm performs very well on a variety of problem types.

7. CONCLUSIONS

The MAGNET automated contracting environment is designed to support negotiation among multiple, heterogeneous, self-interested agents over the distributed execution of complex tasks. The MAGNET testbed is a prototype implementation of a Customer agent, along with a population of simulated Supplier agents. It is highly configurable and extensible, and has been used for several statistical studies aimed at understanding the decision processes for a Customer agent.

The current system has proven to be very useful for the types of statistical studies we have pursued so far. Future plans call for more focus on mixed-initiative interaction, and our current user interface is too primitive to support that work.

Some domains, notably the International Shipping domain we are currently studying in collaboration with North Star Import-Export, will require an enhanced plan representation to deal with the fact that alternate routes or shipping modalities may be acceptable.

A major need in this area of research is the establishment of a set of benchmark problems by which different strategies can be compared. Leyton-Brown et al [13] have proposed a test suite called CATS for testing combinatorial auction systems. It solves part of the problem, but it only deals with bids, not the RFQ, and it does not handle the precedence relations needed in the MAGNET environment.

Acknowledgments

Partial support for this research is gratefully acknowledged from the National Science Foundation under award NSF/IIS-0084202.

8. REFERENCES

[1] A. Andersson, M. Tenhunen, and F. Ygge. Integer programming for combinatorial auction winner

determination. In *Proc. of 4th Int'l Conf on Multi-Agent Systems*, pages 39–46. IEEE Computer Society Press, July 2000.

[2] A. Chavez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. In *Proc. of the First Int'l Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, UK, April 1996.

[3] J. Collins, C. Bilot, M. Gini, and B. Mobasher. Decision processes in agent-based automated contracting. *IEEE Internet Computing*, pages 61–72, March 2001.

[4] J. Collins, R. Sundareswara, M. Gini, and B. Mobasher. Bid selection strategies for multi-agent contracting in the presence of scheduling constraints. In A. Moukas, C. Sierra, and F. Ygge, editors, *Agent Mediated Electronic Commerce II*, volume LNAI1788. Springer-Verlag, 2000.

[5] J. Collins, B. Youngdahl, S. Jamison, B. Mobasher, and M. Gini. A market architecture for multi-agent contracting. In *Proc. of the Second Int'l Conf. on Autonomous Agents*, pages 285–292, May 1998.

[6] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *Int. Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1997.

[7] Y. Fujishjima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proc. of the 16th Joint Conf. on Artificial Intelligence*, 1999.

[8] C. Gomes, B. Selman, N. Crato, and H. Kautz. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *Journal of Automated Reasoning*, 24(1/2):67–100, 1999.

[9] R. H. Guttman, A. G. Moukas, and P. Maes. Agent-mediated electronic commerce: a survey. *Knowledge Engineering Review*, 13(2):143–152, June 1998.

[10] S. Helper. How much has really changed between us manufacturers and their suppliers. *Sloan Management Review*, 32(4):15–28, 1991.

[11] L. Hunsberger and B. J. Grosz. A combinatorial auction for collaborative planning. In *Proc. of 4th Int'l Conf on Multi-Agent Systems*, pages 151–158, Boston, MA, 2000. IEEE Computer Society Press.

[12] P. Langley. Systematic and nonsystematic search strategies. In *Proc. Int'l Conf. on AI Planning Systems*, pages 145–152, College Park, Md, 1992.

[13] K. Leyton-Brown, M. Pearson, and Y. Shoham. Towards a universal test suite for combinatorial auction algorithms. In *Proc. of ACM Conf on Electronic Commerce (EC'00)*, Minneapolis, MN, October 2000.

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

[15] C. R. Reeves. *Modern Heuristic Techniques for Combinatorial Problems.* John Wiley & Sons, New York, NY, 1993.

[16] N. M. Sadeh, D. W. Hildum, D. Kjenstad, and A. Tseng. MASCOT: an agent-based architecture for coordinated mixed-initiative supply chain planning and scheduling. In *Workshop on Agent-Based Decision Support in Managing the Internet-Enabled Supply-Chain, at Agents '99*, pages 133–138, May 1999.

[17] T. Sandholm. An algorithm for winner determination in combinatorial auctions. In *Proc. of the 16th Joint Conf. on Artificial Intelligence*, pages 524–547, 1999.

[18] T. W. Sandholm. *Negotiation Among Self-Interested Computationally Limited Agents.* PhD thesis, Department of Computer Science, University of Massachusetts at Amherst, 1996.

[19] K. Sycara and A. S. Pannu. The RETSINA multiagent system: towards integrating planning, execution, and information gathering. In *Proc. of the Second Int'l Conf. on Autonomous Agents*, pages 350–351, 1998.

[20] M. Tsvetovatyy, M. Gini, B. Mobasher, and Z. Wieckowski. MAGMA: An agent-based virtual market for electronic commerce. *Journal of Applied Artificial Intelligence*, 11(6):501–524, 1997.

[21] W. E. Walsh, M. Wellman, and F. Ygge. Combinatorial auctions for supply chain formation. In *Proc. of ACM Conf on Electronic Commerce (EC'00)*, October 2000.

[22] W. E. Walsh, M. P. Wellman, W. Peter R, and J. K. MacKie-Mason. Some economics of market-based distributed scheduling. In *18th Int'l Conf. on Distributed Computing Systems*, pages 612–621, May 1998.

[23] M. P. Wellman and P. R. Wurman. Market-aware agents for a multiagent world. *Robotics and Autonomous Systems*, 24:115–125, 1998.

[24] P. R. Wurman, M. P. Wellman, and W. E. Walsh. The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In *Second Int'l Conf. on Autonomous Agents*, pages 301–308, May 1998.