Mixed-Initiative Decision Support in Agent-Based Automated Contracting

John Collins, Corey Bilot, Maria Gini
Department of Computer Science and Engineering, University of Minnesota
Bamshad Mobasher
School of Computer Science, Telecommunications, and Information Systems, DePaul University

ABSTRACT

Using principles from Expected Utility Theory, we analyze the criteria that a customer agent in agent-based automated contracting would use in making decisions during the bidding cycle. We use the University of Minnesota's MAGNET automated-contracting environment as a framework for this analysis. Two decisions must be made by a customer agent during this process: deciding the composition of the Request for Quotes, and evaluating and awarding bids. We show how principles from Expected Utility Theory can be applied in a mixed-initiative environment, where user preferences control the decision-making process, and user decisions are final. Finally, we show how the market infrastructure can support agent decision-making by gathering and analyzing statistical data on activities in the market.

1. INTRODUCTION

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 to outside contractors, 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 enable automated contracting among manufacturers, part suppliers, and specialized subcontractors.

Deciding what to outsource and to whom, ensuring that the tasks are done in the proper sequence (parts cannot be painted before they are finished) and that the final product is ready within the time constraints, is currently the job of a human decision-maker. The decision-maker also keeps track of any delays from suppliers that could jeopardize the completion of the tasks, and renegotiates with them or other suppliers as needed.

Most current e-commerce systems rely on auctions, but 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 [11]. These factors must be taken into account while negotiating contracts.

In addition, contracting is only a step in the process of producing goods. When component parts have to be assembled and there are time dependencies among the operations, scheduling becomes a major factor. A schedule with slack between tasks is less risky than a tight schedule, but in made-to-order products speed is the essence and taking extra time might prevent a supplier from getting a contract. After contracts have been awarded, there is one more complication. A late delivery of a component part might produce a cascade of devastating effects on the rest of the contracted work. This has to be considered at the time the contract is negotiated.

The University of Minnesota's MAGNET system [7] is designed to automate this decision making process as much as desired by the decision-maker. We describe an expected-utility approach that makes effective use of the capabilities of a distributed community of agents engaged in negotiation over contracts, by using them to support human decision-making. Agent interactions are mediated through an independent market infrastructure which, among other services, provides a domain ontology, a contracting protocol, authentication services, and tracks the requests, commitments, and progress towards task completion among the agents.

In this paper we focus in particular on two decision processes that take place during the bidding cycle:

1. The first decision is to determine the specific contents of a Request for Quotes (RFQ) at the start of a bidding cycle. This decision determines how much time suppliers are given to submit bids, and it determines an approximate schedule by setting limits on the start and end times for each individual task.
2. At the conclusion of the bidding cycle, the agent must decide whether to award bids, and which bids to award. Evaluation of bids is complicated. Bids have multiple attributes, including cost, duration, and time constraints. Finding the best combination of bids requires not only deciding what to optimize (cost, time, minimize the number of suppliers, give priority to spe-

cific suppliers, satisfy pre-existing contracts, etc) but also assessing risk. Selecting more reliable suppliers or dealing only with prequalified suppliers is a way of reducing risk, but might increase cost.

We expect that fully autonomous behavior will often be impractical or unacceptable because human notions of utility tend to be inconsistent and difficult to model [1], because of risk factors that cannot easily be quantified, or simply because decision making is the responsibility of a person. In some domains, where specifications and business relationships are in place among the negotiating entities, fully autonomous behavior may be acceptable. The level of autonomy of the agent should be adjustable. A mixed-initiative agent should allow a human decision-maker to accept or override recommendations, and it should analyze and present decision criteria clearly and concisely. Our preliminary study sheds some light on how to guide a decision-maker through the important decisions in the bidding cycle.

We start by describing the information content in the interaction between agents during the bidding process. Then we describe how the principles of Expected Utility Theory can be applied to the MAGNET domain. Next we describe the utility-based decision process and its application to the two decision problems listed earlier. Along the way, we make note of the statistical data the agent needs to support its decision processes, data that the MAGNET market can be configured to collect and analyze. Finally, we relate our work to other published work.

2. AGENT INTERACTIONS IN MAGNET

The MAGNET environment is a distributed set of objects that can support electronic commerce in a variety of domains, from the simple buying and selling of goods to situations that require complex multi-agent negotiation and contracting.

Each *Market* within MAGNET is a forum for commerce in a particular business area, and includes a set of domain-specific services, as shown in Figure 1.

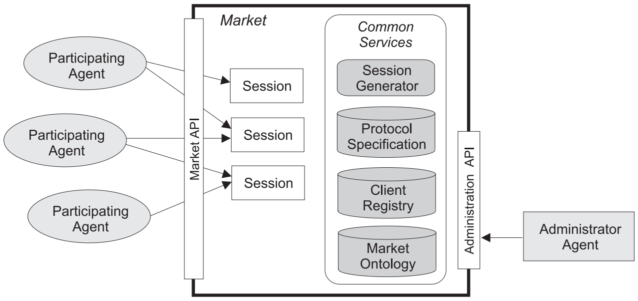

Figure 1: The Structure of a MAGNET Market

An important component of each market is a set of current *Market Sessions* in which the actual agent interactions occur. Each session is initiated by a single agent for a particular purpose, and in general multiple agents may join an existing session as clients. The session enforces the protocol rules, and maintains its internal state according to the protocol activity and the passage of time. The architectural components of MAGNET and some details of its implementation are described in [7].

Within this architecture, an agent has three basic functions: planning, negotiation, and execution monitoring. Within the scope of a negotiation, we distinguish between two agent *roles*, the *Customer* and the *Supplier*. A Customer is an agent who has a goal to satisfy, and needs resources outside its direct control in order to achieve its goal. The goal may have a *value* that varies over time; for present purposes, we assume that value drops to a negligible amount after the goal *deadline*. A Supplier is an agent who has resources and who, in response to a request for quotes, may offer to provide resources or services, for specified prices, over specified time periods.

The market contains an *Ontology* that describes the types of tasks or goods that the market deals in. Each description not only describes the item, but also contains statistics, including details like the number of suppliers that typically will bid on the item, and how long the task typically takes. The market also keeps a *Registry* of suppliers that have expressed an interest in participating in market activities, and maintains performance statistics that customers can use in their decision processes.

The interaction between customer and supplier agents starts with a Request for Quotes (RFQ) issued by the customer, followed by a set of bids submitted by interested suppliers, and concludes with a set of bid awards issued by the customer. After contracts are awarded, the execution phase starts.

For the purpose of this analysis, we are primarily concerned with the decisions the customer must make during the bidding cycle, and we will not consider plan execution,

The exchange of messages between agents is designed to simplify negotiations without loss of generality. It is modeled after the leveled commitment protocol proposed by Sandohlm [19].

- The customer issues an RFQ which includes a specification of each task, and a set of precedence relations among tasks. For each task, a time window is specified giving the earliest time the task can start and the latest time the task can end.
- A supplier's bid includes a price for the task, a portion of the price required to be paid as a non-refundable deposit at the time the bid is awarded, an estimated duration for the task, and a time window within which the task can be started.
- When the customer awards a bid, it must pay the deposit and specify the actual time, within the supplier's specified time window, at which it wishes to begin the task.
- When the supplier completes a task, the customer must pay the remainder of the price, beyond the deposit, as specified in the awarded bid.
- If the supplier fails to complete a task, the price is forfeit and the deposit must be returned to the customer. A penalty may also be levied for non-performance, but we ignore this complication at this point.

Once bids have been awarded, a secondary protocol allows agents to negotiate schedule changes. This avoids outright failure and reduces risk for both parties, at the cost of complicating the behavioral requirements of agents during plan execution.

3. A MOTIVATING EXAMPLE

Assume Acme Widgets asks its agent to find the resources to prepare a display for a trade show in two weeks. Acme's sales department estimates they can book sales during the show that will result in $10 000 in profit (not including the cost of the display), if the display is ready in time.

There are three tasks to be done, and there is some uncertainty in the abilities of the suppliers to deliver on time. We ignore the uncertainty in the profit number. Figure 2 shows the financial situation of the customer agent as the plan progresses. Deposits on all tasks $(d_1 + d_2 + d_3)$ are paid when bids are awarded at the conclusion of the bidding cycle, and payments for each of the tasks, i.e. the agreed cost minus the deposit, $(c_1 - d_1$ etc.) are made as the tasks are completed. Note that if a task n is not completed, then the deposit d_n is returned to the customer. When the plan is complete, the value V of the goal accrues.

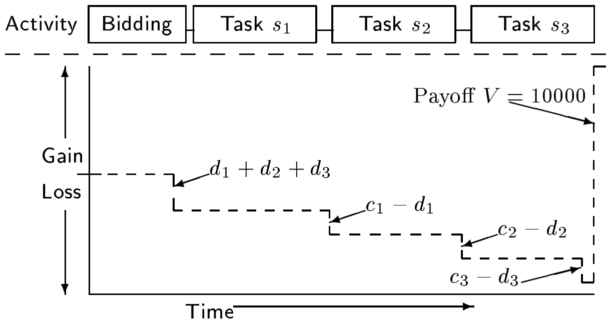

Figure 2: Financial position during project

Assuming that the customer has received multiple bids that specify different costs, deposits, and time parameters, the customer's decision process will attempt to award a combination of bids that maximizes its expected utility at the completion of the project. Because that expected utility involves some probability of loss as well as a probability of gain, we must deal with the risk posture of the person or organization on whose behalf the agent is acting. To do this, we will use the notion of *marginal expected utility*, which is the expected change in a decision maker's overall utility due to some decision.

We are interested in automating (completely or in part) the process of decision making, so we need to specify more precisely what we want to compute before explaining how we expect to perform the computation.

4. EXPECTED UTILITY

If an automated agent is to produce plans that are acceptable to a human decision-maker, our agent must model two concepts: (1) decision-making in an environment of uncertainty, and (2) risk aversion.

To model these concepts, we will use Expected Utility Theory (EUT) [1]. EUT models decision-making under uncertainty by using probabilities and a construct known as a utility curve, $U(W)$, a function that maps a level of "wealth" to a level of utility. According to EUT, a decision-maker who is faced with an opportunity consisting of a set of n wealth-based outcomes will calculate the expected utility over the set of outcomes:

$$E(U) = \sum_{i=1}^{n} U(W_i) p_i \qquad (1)$$

where p_i is the probability of outcome i, and W_i is the resulting wealth of the decision-maker if outcome i is realized. Stated another way, the decision-maker weighs the utility of each outcome within the opportunity. To make her decision, the decision-maker compares this expected utility, $E(U)$, to her current utility, $U(W_0)$, where W_0 represents her current wealth. If the opportunity's expected utility, $E(U)$, exceeds her current utility, $U(W_0)$, she will pursue the opportunity. Similarly, a decision-maker faced with multiple opportunities can decide which (if any) she will pursue by comparing the expected utilities of the opportunities and her current level of utility.

EUT also models the phenomenon of risk aversion. A decision-maker is said to be risk-averse if she will reject an opportunity that has a positive expected value. In effect, the decision-maker is risk-averse if there exists an opportunity that has a positive expected value but an expected utility which is lower than the decision-maker's current level of utility. To model this, EUT stipulates two additional axioms pertaining to the shape of the decision-maker's utility curve: (1) the first derivative of U with respect to W is positive, i.e. a decision-maker always prefers more wealth to less wealth, and (2) the second derivative of U with respect to W is negative, i.e. each successive increment in wealth yields less additional (but still positive) utility.

It is this second axiom that represents risk aversion. To see how this creates risk averseness, consider the following inequality derived from the second axiom.

$$U(W_0) - U(W_0 - X) > U(W_0 + X) - U(W_0) \quad for\ X > 0. \qquad (2)$$

The magnitude of the decrease in utility from losing X always exceeds the magnitude of the increase in utility from winning X, regardless of the initial level of wealth, W_0. In a sense, our decision-maker prefers avoiding losses over seeking gains because losses result in potentially steep decreases in utility in comparison to the increases in utility associated with gains in wealth.

We use EUT to guide the agent in situations in which there is a trade-off between the overall cost of a plan and the likelihood of the plan succeeding. For example, the agent may need to choose between suppliers, some of whom charge a higher price but are more likely to complete the task successfully; others of whom are less likely to complete the task but who will charge less. By computing the expected utility of the scenarios, the agent can choose from among them.

More specifically, in order to compute the customer's expected utility of a plan being executed by a set of supplier agents, we treat the plan as a set of ordered task completion events. Each event has a probability of succeeding, and at the time of each event the customer must pay some supplier. After completion of the last task, the customer gains the benefit of plan completion. If any task fails to complete, we assume the plan is abandoned, and deposits paid to downstream suppliers (suppliers who have not yet begun processing their respective tasks) are forfeited. For n tasks, this gives

$$\begin{aligned} E(U) &= U(W_0) + \sum_{i=1}^{n} \left(M(-z_i)(1-p_i) \prod_{j=1}^{i-1} p_j \right) \\ &\quad + M(V) \prod_{j=1}^{n} p_j \end{aligned} \tag{3}$$

where $M(x)$ is the change in utility due to a financial gain of x, the p_i are the success probabilities of the successive tasks, the z_i are the cumulative "debits" resulting from each task completion (the d_i and c_i of Figure 2) , and V is the net "credit" that accrues on plan completion.

We call the function $M(x)$ the *marginal expected utility* of a gain of x. We introduce this notion to simplify thinking about situations where we are only concerned about changes in wealth due to some decision. Denoting with ΔW_i the change in wealth relative to W_0, for outcome i, Equation 1 becomes

$$E(U) = U(W_0) + \sum_{i=1}^{n} M(\Delta W_i) p_i \tag{4}$$

It is important to understand that, for our purposes, $M(\Delta W)$ is really a qualitative concept, not a function we expect to be able to compute exactly. Many functions have been proposed [8], but little is known about how to elicit preferences from a human user or organization that will yield an accurate utility function. Instead, we recognize its existence and its general shape. To compute values, we will bound $M(\Delta W)$ with a linear function as an upper bound (risk-neutral). This is fairly close to reality for small gambles in any case.

We define successful plan execution as "completed by the deadline," and we define successful completion of a task as "completed without violating temporal constraints in the plan." Note that a task can be completed successfully even if it is not finished within the duration promised by the bidder, as long as the schedule has sufficient slack to absorb the overrun. If a plan is completed after its deadline, it has failed, and we ignore any residual value of completed work to the customer. We plan on extending our analysis to more complicated cases, but we will use these definitions as a starting point.

In the example of Figure 2, the expected utility $E(U)$ for Acme Widgets resulting from the endeavor is:

$$\begin{aligned} E(U) &= U(W_0) + M(-d_2 - d_3)(1-p_1) \\ &\quad + M(-c_1 - d_3)p_1(1-p_2) \\ &\quad + M(-c_1 - c_2)p_1 p_2 (1-p_3) \\ &\quad + M(-c_1 - c_2 - c_3)p_1 p_2 p_3 \\ &\quad + M(V) p_1 p_2 p_3 \end{aligned} \tag{5}$$

where V is the \$10 000 profit, d_n is the non-refundable deposit that must be paid when bid n is awarded, c_n is the price that must be paid when task n is completed, and p_n is the probability that task n will be completed by the deadline agreed to.

To compute the expression in Equation 5, the customer agent needs to estimate, for each task and supplier, the probabilities that the tasks will be completed on time. It then must compute its marginal utility $M(x)$ for each possible outcome. Statistical information about tasks and suppliers is available from the market, but the utility function depends on the customer. A human decision-maker who trusts her agent to make autonomous decisions will specify an analytical form for her own utility function. A decision-maker who prefers to make decisions directly will use the agent to do some computations and present alternatives and will keep to herself the final decision.

If the set of tasks includes potentially parallel activities, the analysis becomes more complex. Different possible schedules may have different marginal utility values, depending on the relative costs and success probabilities of the individual tasks. Once a task starts, the customer is liable for its full cost at completion, regardless of whether in the meantime the plan as a whole has been abandoned due to a failure on some other branch of the plan.

As an example, consider the plan in Figure 3. In this plan, depending on expected task durations, it may be possible to complete task s_2 before starting s_5, or to delay the start of s_2 to after completion of either or both of s_3 and s_4. It may even be possible to serialize s_3 and s_4 in either order if the plan has sufficient slack. Each of these orderings will yield a different value of $E(U)$. For example, if s_2 is expensive relative to tasks s_3 and s_4, then it should be delayed until after both s_3 and s_4 have been completed, if possible. This will reduce the number of terms in Equation 3 in which the cost of s_2 appears.

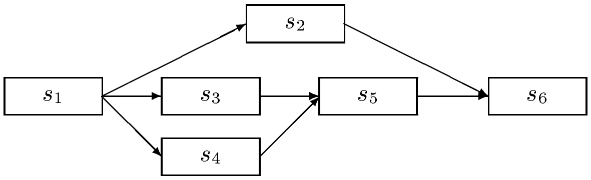

Figure 3: Branching task network

The expected utility model requires knowing the probabilities of the various events. The agent is likely to know (or be able to learn from the market) some of those probabilities better than others.

How sensitive is a risk estimate to uncertainty about the reliabilities of suppliers? One way to answer this is to look at the partial derivatives of $E(U)$ with respect to p_k in Equation 3:

$$\begin{aligned}\frac{\partial E(U)}{\partial p_k} &= -M(-d_k)\prod_{j=1}^{k-1} p_j \\ &+ \sum_{i=k+1}^{n}\left(M(-z_i)(1-p_i)\frac{\prod_{j=1}^{i-1} p_j}{p_k}\right) \qquad (6) \\ &+ M(V)\frac{\prod_{i=1}^{n} p_i}{p_k}.\end{aligned}$$

If the upper bounds of any of the derivatives are negative, then the message is that we are better off abandoning the plan, because a higher probability of success for that element leads to a worse overall outcome. If any of the derivatives are especially large, and the confidence in the corresponding probability estimates are low, then the decision maker should be warned of the uncertainty and its potential impact, and the agent should consider adding slack to the schedule in order to allow for recovery.

Because of the temporal constraints between tasks, failure to accomplish a task does not necessarily mean failure of the goal. Recovery might be possible, provided that whenever a supplier decommits there are other suppliers willing to do the task and that there is sufficient time to recover without invalidating the rest of the schedule. This complicates the selection of which bids to accept. The lowest cost combination of bids and the tightest schedule achievable is not necessarily the preferable schedule because it is more likely to be brittle.

5. DECISION PROCESSES OF THE CUSTOMER AGENT

As described earlier, there are two points in the bidding cycle where the customer agent needs to take utility and risk into account. One is during the composition of the RFQ, where the tasks and time windows are specified. The other is during evaluation of bids, when decisions need to be made regarding which bids to accept. Factors that must be considered include the price, the time window specified in the bid, the reliability of the supplier, and the confidence we have in the reliability data.

5.1 Composing the Request For Quotes

When the customer composes the RFQ, the goal is to maximize the expected marginal utility of the plan at completion time. The customer can't schedule tasks directly; instead, it must issue a RFQ that is likely to garner a set of bids from potential suppliers, that will then be composed into a schedule. There are three time-related factors in the RFQ that can affect the successful outcome:

1. the allocation of time between bidding and execution,
2. the allocation of time within the bidding cycle between suppliers and the customer,
3. the time constraints on each task.

Suppliers need time to evaluate their resource availability and compose bids, and the customer needs time to evaluate bids. If more time is allocated to the bidding process, then the time available for execution will be reduced, and the risk of plan failure increased. If less time is allocated to the bidding process, then either the suppliers, the customer, or both will have less time to consider their options.

We have studied allowing overlaps between the supplier and customer portions of the bidding cycle [3]. The conclusion was that this type of overlap allows suppliers to manipulate the customer by adjusting the timing of their bid submissions, although there may be time-critical situations where the benefits of overlapping bidding and evaluation is worth the extra cost.

How can a customer agent know how much time to allocate to the supplier portion of the bidding process? The customer's principal strategy in allocating time between itself and suppliers is to allocate just enough time to itself to make a decision, and no more. This is because suppliers will likely either not bid, or will raise prices, if they have to reserve resources while speculating on outstanding bids. Also, any extra time spent in customer decision-making reduces the time available for plan execution. We have attempted to characterize our bid-evaluation process [4] in order to provide guidance for this time allocation problem.

The RFQ includes early start and late finish times for each task. Setting these "time windows" is the second major decision the customer needs to make prior to soliciting bids. At the conclusion of the bidding cycle, the agent will need to compose the bids into a feasible schedule. The ability to do that depends on suppliers returning bids that satisfy precedence constraints in the plan. There are two decisions here: the relative allocation of time among the tasks, and the extent to which the time windows of adjacent tasks (connected by precedence relations) are allowed to overlap.

The MAGNET market provides three kinds of data about each task type in its ontology that can be used to make these decisions: (1) the number of bidders that are likely to submit bids, (2) the expected duration, and (3) the amount of variability in the duration data.

To construct the time windows, we construct an initial schedule using the expected duration data, and set the initial time windows using the Critical Path algorithm [12]. The Critical Path algorithm walks the directed graph of tasks and precedence constraints, forward to compute the earliest start times for each task, and then backward from the goal time to compute the latest finish and latest start times for each task. The minimum duration of the entire plan is called the *makespan* of the plan. The difference between the goal time and the latest early finish time is called the *total slack* of the plan.

Then we adjust it based on bidder population and variability data. The detailed relationships between the bidder count and variability data and the optimal adjustments that need to be made in the RFQ bids schedule are still under active investigation. Our current approach is to increase relative time allocation when duration data is more variable, and to increase the overlap when the number of bidders is higher.

There is a tension between issuing a RFQ that will guarantee the feasibility of any plan constructed with the resulting bids, and issuing one that will solicit the maximum number of bids. We assume that supplier's bids result from an evaluation of their current resource commitments, and therefore larger time windows will result in more bids. Suppliers know that more time flexibility in their bids will give them a competitive advantage [5].

5.2 Evaluating and Awarding Bids

Once bids have been received from suppliers, the customer's goal is to find and schedule the combination of bids that maximizes $E(U)$, and award them. The problem is to find a "good" mapping of bids to tasks and then find a schedule for those bids that has a low risk of unrecoverable failure [4]. A "good" mapping is one that *covers* all the tasks, is *feasible* in terms of satisfying the temporal constraints, and is relatively low-cost. Maximizing $E(U)$ at this point is equivalent to finding a set of bids with the combination of cost and risk factors that the user is most comfortable with [6].

MAGNET customer agents incorporate an adaptive anytime search with multiple selectors and evaluators. The algorithm adjusts itself to the problem size by using either a systematic iterative-improvement search, or a simulated-annealing search [15] with adjustable beam width. Evaluators measure attributes of the bid-task mapping such as coverage, feasibility, cost, and a variety of risk factors. Selectors update a mapping by adding and removing bids. A random selector just picks a random bid that is not part of the existing mapping and applies it. A focused selector chooses a bid that is likely to improve some particular attribute of the plan (but may not necessarily preserve other attributes) like cost, coverage, or the amount of schedule slack for a particular task.

Used interactively, the MAGNET search engine is able to produce recommended mappings that optimize different combinations of attributes, and explain to the user what those attributes are. It is also able to take any given mapping as a starting point, fix parts of it under user control, and then search for improvements to specific attributes of that mapping. Figure 4 illustrates improvement curves obtained on one set of experimental data with different selectors [4].

As already mentioned, risk comes from multiple sources: availability of suppliers, supplier reliability, type of task, insufficient time to recover from failures, etc. Different sources of information can be used to estimate risk, such as knowledge accumulated by the agent and knowledge collected by the market. Knowledge of the domain, or experience, could inform the agent that some kinds of work are inherently riskier than others. Examples might be tasks that are more complex or involve more creative content. Experience with suppliers could also lead the agent to ascribe a higher risk to some suppliers than others.

The MAGNET market maintains several types of data on each supplier and on each task type in support of risk evaluation.

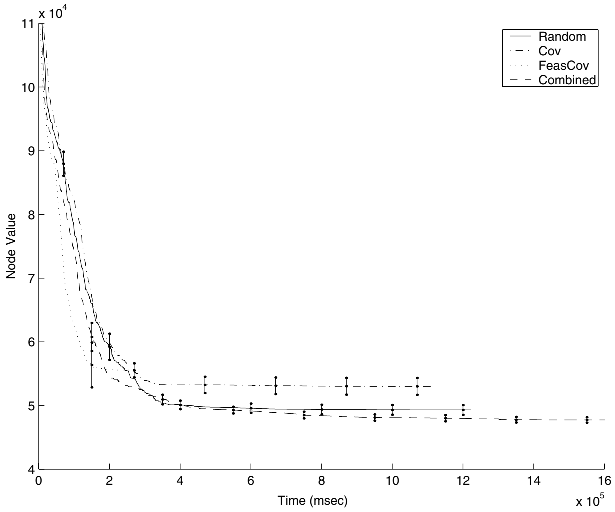

Figure 4: Simulated Annealing Improvement Curves

- *Performance to commitment* P_c – The ratio of successes to attempts, where the task was completed within the promised duration. It does not include bid awards that were abandoned by the customer before the task was started.
- *Performance with overruns* P_l – The proportion of attempts that were completed late.
- *Overrun duration* t_l – The lateness of late completions, with respect to bid durations.

For each of these factors, the market maintains the sample mean, sample size, and variance. This permits agents to compute a confidence interval, in order to be able to make reports to the user of the form "there is a $n\%$ probability that your risk is less than x."

To estimate the risk in a bid-task mapping, we compute a lower bound of the risk R. This is the absolute value of the negative part of the expected value computation, not including the payoff for plan completion:

$$R = \left| \sum_{i=1}^{n} \left(-z_i(1-p_i) \prod_{j=1}^{i-1} p_j \right) \right| \tag{7}$$

These risk estimates cannot be produced without a fixed schedule. This is because specific start times determine the schedule slack available for recovery if a supplier misses a deadline, and because they affect the ordering of parallel tasks.

Clearly, minimizing risk by adjusting the start times of tasks is a non-linear combinatorial optimization problem, since the individual completion probabilities can be influenced by the amount of slack available to recover from failure. Different approximations can be used, such as:

1. Allow the user to choose a confidence level, and use that to compute completion probabilities from market data on the individual suppliers and tasks.

2. Estimate marginal completion probabilities given additional time. We will use the performance with overrun and overrun duration data, and assume that the improvement in completion probability is linear in time as shown in Figure 5.
3. Initialize the start times of each task to be as early as possible, consistent with precedence constraints and bid specifications. This is a heuristic driven by the observation that later tasks tend to be riskier because of the larger outlays later in the plan.
4. Use a greedy optimization technique such as Joslin and Clements' "Squeaky Wheel" method [13] to reduce the risk as computed by Equation 7.
5. Present the user with the choices, the risk data, and the sensitivities from Equation 6.

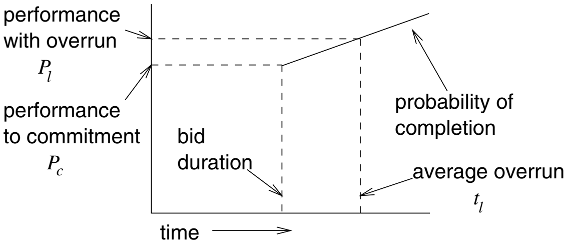

Figure 5: Estimating task completion probability

Risk can also be reduced by consolidating tasks with single suppliers. Suppliers can bid on "packages" composed of subsets of tasks from the RFQ. In general, the customer is better off from a risk standpoint if it takes these packages, assuming that the supplier is willing to be paid for the whole package at the time of its completion. In some cases, the customer may be willing to pay a premium over the individual task prices in order to reduce risk. The advantage is greater toward the end of the plan than near the beginning. To see why this is so, we restructure Equation 5 to consolidate tasks 2 and 3, without changing the costs and deposit amounts:

$$\begin{aligned} E(U) &= U(W_0) + M(-d_2 - d_3)(1 - p_1) \\ &\quad + M(-c_1)p_1(1 - p_{23}) \\ &\quad + M(-c_1 - c_2 - c_3 + V)p_1 p_{23}. \end{aligned} \tag{8}$$

Notice that the fourth term from Equation 5 is missing, and the third term represents a smaller outlay. Also, the probability factor in the last term may be larger, assuming that the probability of the supplier delivering on the consolidated task (p_{23}) is nearly the same as delivering on a single task (p_2 or p_3).

As the search for "good" mappings progresses, we use the principle of *dominance* from Multi-Attribute Utility Theory [22] to determine which solutions are worth scheduling, and ultimately which solutions to present to the user for modification and approval. Attributes can include cost, estimated risk, number of suppliers, expected completion time, and others such as the use of a particular supplier, at the option of the user. Solutions are added to or remain in the list of candidate solutions only if they are not dominated by other solutions in the list.

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, 16, 20, 23] and our own MAGMA architecture [21]). Most market architectures limit the interactions of agents to manual negotiations, direct agent-to-agent negotiation [19, 9], or various types of auctions [24].

Auctions are becoming the predominant mechanism for agent-mediated electronic commerce [10]. AuctionBot [24] and eMEDIATOR [18] are among the most well known examples of multi-agent auction systems. They use economics principles to model the interactions of multiple agents. Auctions are not always the most appropriate mechanism for the business-to-business transactions we are interested in, where convenience of scheduling, reputation, and maintaining long term business relations are often more important than cost.

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.

Most work in supply-chain management is limited to strict hierarchical modeling of the decision making process, which is inadequate for distributed supply-chains. Each organization in the supply-chain has its own set of objectives, and should be considered as self-interested or even antagonistic as opposed to cooperative. A notable exception is the MASCOT [17] agent-based system. The major difference with our proposed work is that agents in MASCOT 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 [14]. Our objective is to negotiate contracts with suppliers that optimize customer's utility.

7. CONCLUSIONS AND FUTURE WORK

The MAGNET automated contracting environment is designed to support negotiation among multiple, heterogeneous, self-interested agents over the distributed execution of complex tasks. If such a system is to be used to augment human decision-making, it must deal with the realities of human notions of marginal utility. We have shown how a marginal-utility approach can be used to support the two primary decisions that must be made by a customer agent during the bidding cycle. We have also discussed the types of statistical data the market infrastructure needs to build and maintain in order to provide the probability and timing data the agents will need for these decision processes.

The MAGNET system has been under development since early 1998. The distributed market infrastructure, including much of the data-collecting capability described here, is in place. The principal elements of a customer agent are complete, including a highly adaptable search engine that is designed to support mixed-initiative bid evaluation in this environment.

Current efforts include a user interface that will present risk information and allow a user to interact with and override agent recommendations, and use the search engine interactively. We are also in the process of extending our simulation environment to enable the full range of behaviors described in this paper to be tested.

Additional study is needed to develop detailed strategies for setting time windows in the call-for-bids, and to relax some of the assumptions used in this analysis. This promises to be a challenging area for both formal and empirical study. There is also a need to ground this work with real-world examples and data. Given such data, a valuable study would be to compare the performance of human decision-makers with and without the support of a MAGNET agent system.

8. REFERENCES

[1] T. Biswas. *Decision-Making Under Uncertainty.* St. Martin's Press, Inc., 1997.

[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, S. Jamison, M. Gini, and B. Mobasher. Temporal strategies in a multi-agent contracting protocol. In *AAAI-97 Workshop on AI in Electronic Commerce*, July 1997.

[4] J. Collins, R. Sundareswara, M. Tsvetovat, M. Gini, and B. Mobasher. Search strategies for bid selection in multi-agent contracting. In *IJCAI Workshop on Agent-Mediated Electronic Commerce*, Stockholm, Sweden, July 1999.

[5] J. Collins, M. Tsvetovat, C. Bilot, R. Sundareswara, T. Lee, M. Gini, and B. Mobasher. A framework for mixed initiative agent-based contracting. In *First IAC Workshop on Internet Based Negotiation Technologies*, March 1999.

[6] J. Collins, M. Tsvetovat, R. Sundareswara, J. V. Tonder, M. Gini, and B. Mobasher. Evaluating risk: Flexibility and feasibility in multi-agent contracting. In *Proc. of the Third Int'l Conf. on Autonomous Agents*, May 1999.

[7] 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.

[8] L. Eeckhoudt and C. Gollier. *Risk Evaluation, Management, and Sharing.* Harvester Wheatsheaf, Hemel Hempstead, Hertfordshire, UK, 1995.

[9] 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.

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

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

[12] F. S. Hillier and G. J. Lieberman. *Introduction to Operations Research.* McGraw-Hill, 1990.

[13] D. E. Joslin and D. P. Clements. "Squeaky wheel" optimization. *Journal of Artificial Intelligence Research*, 10:353–373, 1999.

[14] D. Kjenstad. *Coordinated Supply Chain Scheduling.* PhD thesis, Dept of Production and Quality Engineering, Norvegian University of Science and Technology, Trondheim, Norway, 1998.

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

[16] J. A. Rodriguez, P. Noriega, C. Sierra, and J. Padget. FM96.5 - a Java-based electronic auction house. In *Second Int'l Conf on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'97)*, London, April 1997.

[17] 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*, May 1999.

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

[19] T. W. Sandholm. *Negotiation Among Self-Interested Computationally Limited Agents.* PhD thesis, University of Massachusetts, 1996.

[20] 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.

[21] 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.

[22] M. P. Wellman. Reasoning about preference models. M.S. thesis MIT/LCS/TR-340, Laboratory for Computer Science, MIT, 1985.

[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*, May 1998.