A Decision-Support System for Quote Generation

Richard Goodwin, Rama Akkiraju, Fred Wu

IBM T. J. Watson Research Center
Yorktown Heights, NY
{rgoodwin, akkiraju, fywu}@us.ibm.com

Abstract

In this paper, we describe a prototype agent-based decision-support system for helping suppliers respond to requests for quote in a business-to-business supply chain. The system provides suggested ways of fulfilling requests and shows alternatives that illustrate tradeoffs in quality, cost and timelines, which allows the decision maker to consider alternatives that reduce cost and improve customer value. The system is implemented in Java and we use examples from paper manufacturing to illustrate the features of our system. In on going work, we are enhancing the prototype to include probabilistic reasoning techniques so that it can create conditional plans that maximize expected utility, subject to the risk preferences of the decision maker. We are also exploring the use of data mining techniques to infer customer preferences and to estimate the probability of winning an order, with a given quote.

Introduction

The proliferation of e-commerce Websites and the growth of business-to-business e-commerce has made it easier for sellers to reach larger numbers of potential buyers of their products, and for buyers to reach larger numbers of potential suppliers of products. These changes are affecting the market for goods and services exchanged in direct procurement, that is, goods that are used in the manufacture of the enterprise's finished products, as well as those exchanged in indirect procurement, those goods that are used in maintenance, repairs, and operations of the enterprise. Unlike retailers, most businesses in the supply chain do not generally sell at fixed prices, but instead negotiate terms with each customer, either for a one-time purchase or a long-term contract. This negotiation process can be time consuming and labor intensive.

Auctions are a long-standing mechanism by which price negotiation is sometimes conducted. In a forward auction, a seller is offering goods or services to a number of buyers, and the auction mechanism is used to determine the price. However, the concept of an auction can be broadened in several ways. First, an auction can be used not only by a seller to sell his products, but also by a buyer to procure needed products; this is commonly referred to as 'RFQ' (Request for Quote) or 'reverse auction.' Second, commodities can be exchanged between multiple sellers and buyers, as in a stock exchange; this is known as a double auction. Third, an auction can be used to negotiate more than price; such multi-parameter auctions could also determine terms of sale such as delivery date, payment terms, and product features. In this paper, although we describe a system for use in quote/bid generation for RFQs, the system and techniques are equally applicable to the analogous cases of forward, reverse, double-sided, combinatorial, and multi-parameter auctions.

As e-commerce sites make it easier to create and run auctions, the number of potential auctions open to each bidder can grow enormously. This can have the effect of overwhelming the persons responsible for bidding in auctions. The bidders need to consider each auction independently for the effect it can have on a number of business objectives, such as production efficiency, profitability, and customer satisfaction. Furthermore, they need to consider auctions collectively because winning a large percentage of them affects the business objectives as compared with winning a small percentage of them. For example, when business is strong and the rate at which auctions are offered is low, a direct procurement buyer may wish to win a large fraction of the items it bids on, and may be willing to quote higher prices to achieve those wins. On the other hand, when the volume of auctions is large, the bidder may not wish to win most of the bids because they would exceed the capacity of his enterprise to consume the product, perhaps leading to excess raw material inventory problems. Similarly, a general contractor bidding in reverse auctions to provide construction services needs to win enough auctions to employ all the tradesmen working for the contractor, but not so many that the jobs cannot be completed on time. How to bid in one auction should depend on outstanding bids in other auctions and intended bids in future auctions. Furthermore, the intention to bid in a future auction may be conditional on the outcome of a previous auction.

In this paper, we focus on the problem of quote generation for RFQs, and describe a decision-support system that aids in generating and evaluating such quotes. This agent-based system makes use of information about production schedules, inventory and availability of goods through electronic marketplaces to recommend good quotes. The current system can quickly consider large

---

numbers of auctions in the context of the current business conditions and recommend sets of quotes in sets of auctions, to relieve the bidder of the burden of performing numerous what-if analyses. This allows the bidder to focus on pricing and non-tangibles, like customer relationship building. We are also extending the system to make use of data mining techniques to elicit buyer preferences about tradeoffs in quote attributes, such as product quality, delivery date and substitutions. We are combining this information with probabilistic reasoning techniques to suggest pricing and bidding strategies that maximize expected profit, subject to the bidders' risk preferences.

Our Approach

To address the problem of quote generation, we are building on our work in production scheduling (Murthy et al. 1999) and promotion creation (Keskinocak et al. 1999). In earlier work, we used an agent based architecture to match supply with demand to create a set of schedules that maximize a set of objectives. In the case of production scheduling, we match orders with production capacity and inventory. We allow for inventory at various stages of production and we allow for reworking of inventory to meet the requirements of an order. In our promotions system, we matched excess inventory from multiple suppliers with expected customer demand, based on salesperson's knowledge of the customer's operation. The resulting matches were presented to customers when the cost of the goods, plus a profit markup, were well below current market prices. This allowed the supplier to offer these items as special deals, tailored to the individual customer.

For quote generation, we have extended the system to pull information on demand and supply from new sources using new integration methods based on web services. In addition, we have enhanced the model to generate quotes for groups of RFQs and not to match each demand in isolation as was done in the promotion generation system. In the promotion system, promotions were offered on a first come first serve basis and if one supply item were offered to multiple customers, the first to respond would have the first option to buy it. In an RFQ setting, there are some significant differences: In some cases, quotes are binding, or withdrawing a quote can reflect negatively on the supplier. There is also a lag in time between when quotes are submitted and orders are awarded, causing uncertainty in availability of resources for making new quotes. Finally, the combination of quotes that are won affects production efficiency, and therefore the cost and the quote price needed to return a profit. The price of the quote in turn affects the likelihood of winning the order, causing a circular dependency.

In our approach to providing decision support for quote generation, we divide the problem into three stages: a demand filtering stage, a supply search stage and a matching stage. In the demand filtering stage, the system retrieves the details of RFQs that the decision maker could respond to. These RFQs are filtered to select only those RFQs for goods and services that the bidder could potentially supply. For example, a paper manufacturer would not want to consider public RFQs for printer's ink, but would entertain RFQs for paper from the same set of printers. The search stage consists of formulating search queries loosely based on the attributes of the input set of RFQs and dispatching them to various sources to obtain supply items that might potentially match the given RFQs' requirements. The search queries are formulated to retrieve not only exact matches for the requested items (finished goods), but also work-in-process or raw materials that the manufacturer can use to produce the requested items. Once the demand and supply items are obtained, the decision-support engine employs a team of agents to generate and evaluate a set of possible solutions that describe how each RFQ could be fulfilled, either in isolation or in combination with other RFQs (Rachlin et al. 1999). The decision support system evaluates the solutions based on a configurable set of evaluation criteria, such as cost, quality of match and delivery date. Finally, a non-dominated set of solutions is presented to the user along with their evaluations. A solution becomes a quote when the decision maker selects it and commits it to the RFQ/Quote system.

System Overview

Figure 1 shows our decision support system for quote generation. It consists of a user interface, a set of RFQ retriever agents, a set of search agents and a decision support engine.

User Interface: A decision maker can use the graphical user interface (GUI) to initiate the quote generation process, step 2 in figure 1. Once the decision support engine generates a set of solutions, user can then view the suggested quotes and their evaluations, together with the supply and demand items used to generate the solutions. The decision maker is then free to drill down into the details, modify the solutions and generate new ones and have these changes evaluated (step13). When the decision maker is satisfied, the chosen quotes are submitted to the RFQ system, step 14.

RFQ retriever agents: RFQ retriever agents retrieve RFQs from one or more RFQ/Quote systems, steps 4 and 5 in figure 1. These systems can include web sites run by the supplier, sites run by the buyer and sites run by a neutral third party market maker. The RFQ retrieval agents not only handle the technical aspects of communication with the RFQ/Quote system but also provide various filters that can be used to focus attention on particular types of RFQs. For instance, a decision maker might prefer to attend to urgent RFQ's first by picking only the ones that are due in a week's time or a user might want to focus on preferred customers' inquiries or on inquiries for specific product types. The filters on the retriever agents are configured by the settings in the user interface that the decision maker selects.

Search agents: The primary task of search agents is to find supply items that are 'potential matches' for the RFQs retrieved by the retriever agents, steps 6 to 10 in figure 1. Search agents come in two types: generic search agents and domain-aware search agents.

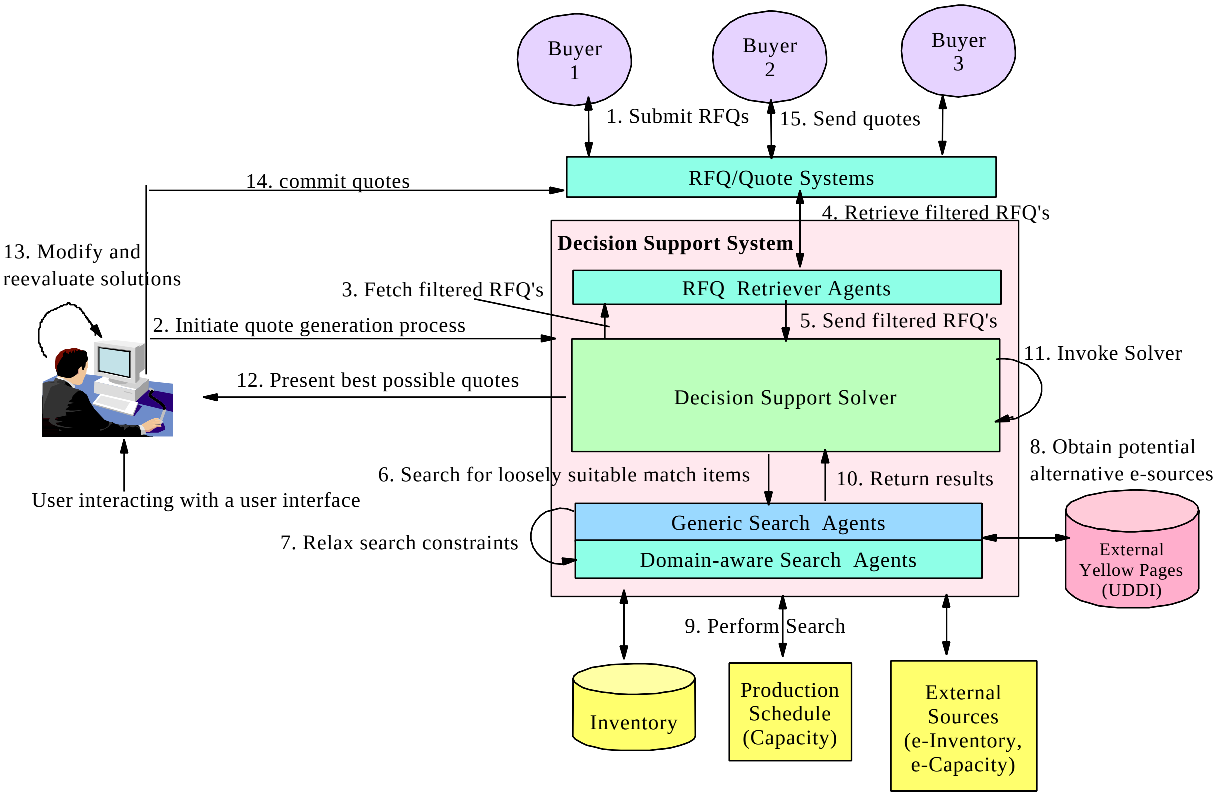

Figure 1: Steps in generating quotes for RFQs using our decision support approach.

Generic search agents: These agents are vested with general knowledge of the sources of supply items. These sources include not only the inventory management, and the production scheduling systems of the enterprise but also external electronic sources such as e-marketplaces. While the location and access details of the enterprise systems can be specified as part of initial configuration, information about external sources can be more dynamic in nature. These agents can obtain information on external sources from public yellow pages directories, such as UDDI (Universal Description, Discovery and Integration) directories. Generic search agents invoke the search services of external sources via Web Services, a set of open standards that facilitate program-to-program interaction by specifying a programmatic means to describe, publish, discover and bind application interfaces.

Domain-aware search agents: While generic search agents handle the technical aspects of communication with the sources of supply items, domain-aware agents deal primarily with reframing the search query to be dispatched to the destinations in order to obtain suitable matches for RFQs based on domain knowledge. For instance, domain-aware search agents have the knowledge of what attributes of an RFQ item can be relaxed while staying within the acceptable parameter ranges. They also have the knowledge of which attributes of a supply item can be modified through reworking and which kinds of machines can be used to do the rework. For example, our paper manufacturing domain agents know that rolls of paper can be trimmed to reduce their width, but can not be made wider, and they know that the diameter of a roll can be decreased by cutting or increased by rolling two rolls onto one spindle. Using such domain knowledge, domain-aware search agents relax the item attribute constraints of a search query to obtain supply items that could include transformable and substitutable items as search results. This relaxed search query, set up by the search agents, results in a set of supply items that gives the decision support system greater flexibility to explore best matches by evaluating the tradeoffs among quality mismatches, price and delivery dates.

Decision support engine: Once the RFQs and the corresponding supply items are obtained, the decision support engine employs a set of agents to generate and evaluate a set of non-dominated solutions. A given supply item could be matched with more than one RFQ item. These combinations yield interesting tradeoffs in the evaluations, namely cost, delivery date, quality, and schedule disruptions. In some cases, the supply items that could be matched with each RFQ form disjoint sets. In such a case, quotes for each RFQ can be generated separately, without a need to consider combinations of quotes or how winning one quote would affect the resources available to supply another. In other cases, a supply item could be used to satisfy more than one RFQ. In such cases, the system considers combinations of quotes and the problem becomes much more interesting. Within the decision support engine, there is a set of agents that generate and improve a population of solutions. Each agent selects which problems and/or partial solutions to work on and employs an algorithm to generate new solutions. This matching problem can be approximated as instances of standard problems such as knapsack problem, bin packing, minimum cost flow etc. and the agents embody many of the standard techniques for addressing these standard problems. Depending on the size of the problem, we also employ mathematical programming techniques such as integer programming. Employing a combination of exact techniques and iterative improvement heuristics techniques results in a robust system that generates high quality solutions that illustrate tradeoffs in cost, tardiness, quality and manufacturing disruptions. More details about the algorithms were discussed in our earlier work on decision support for paper industry (Murthy et al. 1999).

Example

To illustrate how the system works and to show how the system assists the decision maker in placing quotes, we provide the following example of a paper converter creating quotes for rolls of paper. We start with some background about a fictional paper converter. Converters are value-add manufacturers that purchase large rolls of paper and cut them into smaller rolls as required by their customers' printing presses or other equipment. We simplify the problem by making a number of assumptions. Of the numerous relevant physical characteristics of paper, we consider only grade and roll width. Our sample problem deals with two grades of paper, A and B, in which grade B can be substituted for grade A, with a slight loss of quality. The converter has two machines, M1 and M2, which can cut wide rolls into narrower rolls; we assume that all rolls have the same diameter, such that each 1000 mm of roll width weighs 1 ton. Each machine has a limit on the width of the roll it can handle; M2 cannot be used on the larger rolls. The machines are committed to other orders until certain dates, after which their time is available. We consider the cost of the raw materials and the use of the machines, but ignore setup time, conversion time, shipping time and shipping costs.

Item	Grade	Width	Quantity (Rolls)	Due Date
D1	A	400	25	2002-04-
D2	A	600	30	012002-04-01

Figure 2: RFQs representing demand.

Figure 2 shows two RFQs, one for 25 rolls of 400 mm wide grade A paper to be delivered on April 1 and the other for 30 rolls of 600 mm wide paper on the same delivery date. Figure 3 shows three types of available resources: inventory, production capacity and e-inventory available from a wholesale e-marketplace.

Supply (Inventory) Items:

Item	Grade	Width	Quantity (Rolls)	Cost per Ton
S1	A	2300	10	80
S2	A	2500	5	90
S3	B	1900	12	60

Electronic Inventory Items:

Item	Grade	Width	Quantity (Rolls)	Date Available	Cost per Ton
E1	A	600	10	03-15	100
E2	B	1500	30	03-01	80
E3	A	900	30	03-15	105

Production Capacity:

Machine	Max. Width	Date Available	Cost per Ton
M1	2500	04-06	15
M2	2000	03-15	10

Figure 3: Three types of resources: inventory, production capacity and e-inventory.

The task for the decision maker is to decide which RFQs to bid on and how much to quote for each one. To assist in this process, the decision support system generates suggestions for how to fulfill each RFQ with the available supply and evaluates each solution in terms of cost, tardiness, quality and manufacturing disruptions. Cost is a measure of the cost of acquiring the material and of machine time for any necessary conversion. Tardiness is measured in ton-days late. Since partial deliveries are generally allowed, this is a measure of the quantity that is late times the number of days that it is late. For comparing the solutions, quality is measured as the number of tons of grade B paper that is substituted for the grade A paper requested. Manufacturing disruptions are measured as the number of tons of paper production in the committed schedule that must be displaced in time in order to produce the new item before the due date. All four objectives (Tardiness, Quality exceptions, Cost, and Disruptions) are to be minimized.

In running the decision support system, the decision maker can choose to create solutions that satisfy any or all of the selected RFQs. Solutions generated to satisfy multiple RFQs are premised on winning all those RFQs; therefore those solutions do not commit any single resource to more than one RFQ. On the other hand, solutions designed to satisfy a single RFQ are created without regard for other RFQs, and are free to utilize all resources available. If the decision maker chooses to examine both types of solutions, (s)he can make contingency plans in case an auction has an unexpected result, i.e. (s)he wins more or less than expected.

Sol	Supply Item	Supply Rolls	Demand Item	Demand Rolls	Machine	Machine Date	T	Q	C	D
1	S2	5	D1	30	M1	04-06	50	0	1312	0
2	S2	5	D1	30	M1	03-30	0	0	1312	12
3	S3	7	D1	28	M2	03-15	0	10	931	0
4	S1	5	D1	25	M1	04-06	50	0	1092	0
5	E3	13	D1	26	M2	03-15	0	0	1345	0
6	S1	10	D2	30	M1	04-06	90	0	2185	0
7	S1	10	D2	30	M1	03-30	0	0	2185	23
8	S2 S1	5 4	D2 D2	20 12	M1 M1	04-06 04-06	90	0	2186	0
9	S3	10	D2	30	M2	03-15	0	18	1330	0
10	S2 E1	5 10	D2 D2	20 10	M1	04-06	60	0	2312	0
11	S2 E2	5 5	D2 D2	20 10	M1 M2	04-06 03-15	60	6	2025	0
12	E3	30	D2	30	M2	03-15	0	0	3105	0
13	S1 S2	10 3	D2/D1 D1	30/10 18	M1 M1	04-06 04-06	140	0	2972	0
14	S1 S3	10 4	D2/D1 D1	30/10 16	M1 M2	04-06 03-15	110	6	2717	0
15	S2 S1 S1 E1	5 4 3 3	D2 D2/D1 D2/D1 D1	20 4/16 6/6 3	M1 M1 M1 M2	04-06 04-06 04-06 03-15	134	0	3040	0
16	S1 S2	10 3	D2/D1 D1	30/10 18	M1 M1	03-15 04-06	30	0	2972	23
17	S3 S1	10 5	D2 D1	30 25	M2 M1	03-15 04-06	50	18	2422	0
18	E3 E1 S2	13 10 5	D1 D2 D2	26 10 20	M2 M1	03-15 04-06	60	0	3258	0
19	S3 S3 E2	10 2 6	D2 D1 D1	30 8 18	M2 M2 M2	03-15 03-15 03-15	0	28	2406	0
20	E3 E1 E3 S1	13 10 17 1	D1 D2 D2 D2	26 10 17 1	M2 M2 M1	03-15 03-15 03-30	0	0	3923	2
21	S1 S2	10 3	D2/D1 D1	30/10 18	M1 M1	03-30 03-30	0	0	2972	30
22	E1 S2 S1	10 5 5	D2 D2/D1 D2/D1	10 10/15 10/10	 M1 M1	 03-30 03-30	0	0	3005	24

Figure 4: Possible solutions. Solutions 1 and 8 are dominated and would not be shown to the decision maker.

In the context of our example, by using the system to generate solutions that satisfy only D1, solutions that satisfy only D2, and solutions that satisfy both D1 and D2, the user can observe the tradeoffs between the various ways to produce each item singly, and the potential benefits that accrue from winning both.

Figure 4 shows a set of 22 possible solutions for this example. The first 5 solutions produce D1 independently.

For example, solution 1 prescribes consuming 5 rolls of the supply item S2 from inventory, to produce 30 rolls of D1, using time on machine M1 to cut each 2500 mm wide roll into six 400 mm wide rolls, with 100 mm of waste. The product will be scheduled for machine M1 on April 6. The evaluations of the solutions appear in the four rightmost columns. Since the RFQ quantity is 10 tons and production will be 5 days late, the tardiness is 50 ton-days. Since the product will be made from grade A paper, there are no quality exceptions. The cost is computed as the weight of the 5 rolls of S2 (12.5 tons) times the sum of the per ton material cost and the per ton conversion cost on machine M1 (90 + 15), yielding 1312.5. There are no disruptions because the production is scheduled after the machine availability date.

Among solutions 1 - 5, solution 5 avoids tardiness, quality problems, and disruptions entirely, but at a high cost. In solution 2, the schedule of M1 is modified to produce the D1 items on time. This results in zero tardiness, but 12 tons of schedule disruption. Solution 3 minimizes the cost, but sacrifices quality. Solution 4 has the same tardiness, quality, and disruption evaluations as solution 1, but lower cost; consequently it dominates solution 1. Our system would therefore not recommend (or display) solution 1 to the user.

Among solutions 6 - 12, which produce D2 independently, only solution 8 is dominated. Solutions 7 and 12 show the tradeoff between disruptions and cost. Solutions 10 and 11, and solutions 9 and 12 show two different tradeoffs between cost and quality. Note that solutions 8, 10, and 11 each use two different supply items to produce the D2 items. Also note that in solution 10, supply item E1 can be used without any conversion to satisfy the RFQ.

Solutions 13 - 22 produce both D1 and D2. With multiple RFQ items, the number of good solutions grows rapidly; all 10 solutions shown are non-dominated. Solution 13 illustrates the possibility of combining D1 and D2 in the consumption of rolls of one supply item, S1. This reduces waste, as can be seen by comparing the evaluations of solution 13 with the combined evaluations of solutions 1 and 6.

As mentioned above, by examining solutions that make RFQ items singly and in groups, the decision maker can evaluate the risks of winning more or fewer auctions than expected. Knowing the evaluations of the contingency plans, the user can adjust the aggressiveness of his/her quotes. For example, if the solutions for making two RFQ items individually evaluate poorly compared to the solutions that make them together, the user might quote more conservatively to increase the likelihood of winning both.

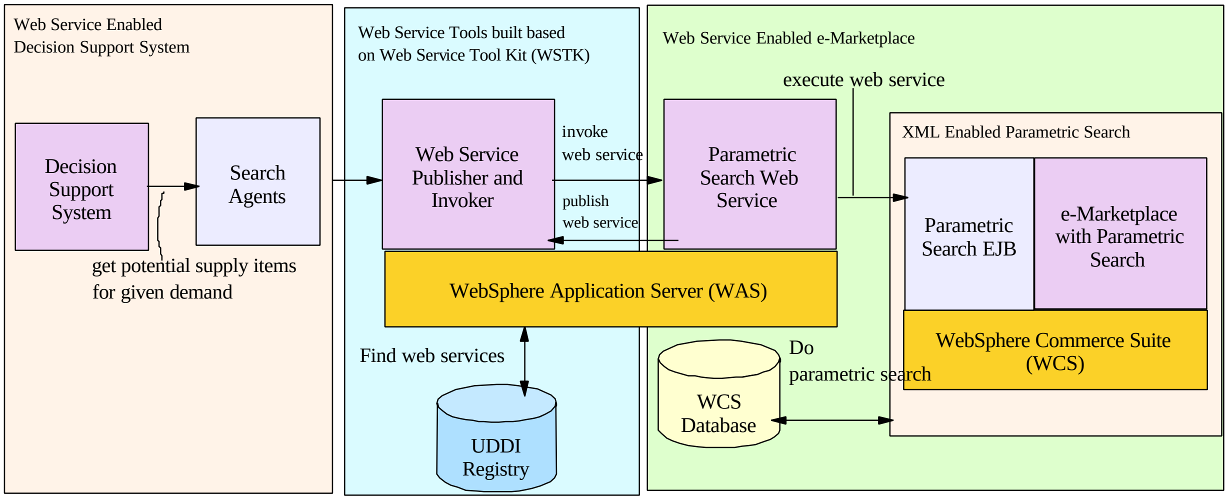

Figure 5: Overview of software component architecture.

Implementation

Our decision support system, implemented in Java, makes use of emerging standards-based technologies such as Web Services (Clement 2002), UDDI (Ariba, IBM, Microsoft 2000), WSDL (Ogbuji 2000) SOAP (Box et al. 2000) and IBM's products, including WebSphere™ Application Server (IBM 2002), and WebSphere Commerce Suite Marketplace Edition™ (IBM 2000). To simulate external supply and demand sources, we created an instance of an e-marketplace with B2B catalog based sales model, that provides RFQs, reverse auctions, forward auctions and double sided auctions. We then made the parametric catalog search facility and the RFQ facilities of the e-marketplace available as web services and published them in a UDDI registry. Using IBM Web Services Tool Kit, WSTK 2.2, (IBM 2001) infrastructure, we web-service enabled the decision support engine. This includes setting up the infrastructure required to find service providers (e-marketplaces) that provide parametric catalog search and RFQ services, to bind to the services once found and to invoke them when required. Figure 5 summarizes the architecture of our system.

Current Status and Future Work

We currently have a working prototype of the system running in our labs at the IBM T. J. Watson Research Center in Yorktown Height, New York. The data used in this prototype was derived from data gathered from customers and used for testing our promotions system. It reflects the characteristics and distribution of attributes requested by real customers of paper manufactures. We plan to pilot deployment of the system with potential customers in the near future.

In addition, we are working to extend the system to add probabilistic reasoning and data mining techniques to enhance the functionality of the system. We have been extending our work on using Markov models to model auctions (Liu, Goodwin, Koenig 2001) and have enhanced our solution language to include probabilistic events and conditional plans. This allows us to generate probability distributions over cost and delivery dates when considering multiple RFQs with non-disjoint sources of supply. Probabilistic models are also required when generating bid to buy goods when there are multiple auctions or other sources of supply. One needs to consider the probability of acquiring a good for a particular cost in an auction versus the cost of manufacturing it or buying it under a fixed price contract.

The use of probabilistic reasoning requires probability estimates. We have already used market simulation to estimate the probability of winning an auction given a particular bid. However, the simulation models require numerous parameters. One approach we are considering is to use data mining techniques to learn the model parameters to improve estimates. Another approach is to try to directly estimate the required probabilities from the results of previous auctions. With estimates for the probabilities of winning with a given bid, the system can suggest bid prices that maximize expected utility, taking into account the bidder's risk preferences.

We are also looking at using data mining techniques to extract buyer's preferences and willingness to accept various kinds of substitutes. For example, it would be useful to determine if a customer is typically more concerned with quality, cost, substitutions or delivery date. Knowing the relative importance can help suggest which options to offer the customer to maximize their satisfaction and maximize the profits of the seller.

Related Work

The company Perfect.com (Perfect 2002) has identified the need for e-marketplace systems that assist on-line suppliers who are receiving large volumes of RFQs. The Perfect.com system offers several mechanisms to reduce the amount of manual work imposed by RFQs. The simple mechanism is a filter that discards those RFQs whose attributes are not sufficiently close to those that the supplier defines as acceptable. The attributes would likely include purchase terms and product characteristics. The filter therefore reduces the number of RFQs that the supplier must consider. The other mechanism offered by Perfect.com is an automated bidding engine that applies preferences provided by the buyer and cost information provided by the supplier to generate quotes automatically. These quotes use the specified preferences and costs to modify the attributes of the product offered from the original RFQ, if such modifications improve the product's value to the buyer. Both mechanisms are implemented in the Perfect.com marketplace system.

The filter mechanism offered by the Perfect.com system is both useful and likely to be acceptable to both buyers and suppliers because it can function without requiring buyers and suppliers to reveal detailed private information. Filter criteria such as "we want only blue widgets" or "we make only yellow widgets" may be considered public and could therefore be used for automatic filtering by the e-marketplace. It is similar to the filter mechanism that our system offers, with the exception that our system allows the filter to be refined per session.

On the other hand, many buyers and suppliers will probably be reluctant to reveal to a Perfect.com-like e-marketplace the detailed private information needed for Perfect.com's system to generate automatic quotes. Such private information as "a blue widget is worth $40 dollars more to me than a yellow widget" or "it costs me $30 dollars more to make a blue widget than a yellow widget" may be too sensitive and dynamic for users to reveal to others. Even though Perfect.com may not permit buyers to view a supplier's cost structure directly, buyers could infer this information by submitting multiple RFQs with systematically varied parameters. Furthermore, some of the criteria to be considered in responding to an RFQ are not static, but depend on the current production environment of the supplier. Accessing such data would require that the e-marketplace have tight integration with the supplier's information systems, which may not be acceptable to the supplier from both privacy and cost standpoints. In contrast, our system can reside within the suppliers security domain, to give it easy and secure access to private data and allow suppliers to deal with multiple sources of demand and supply without revealing information about which suppliers it is considering.

Maxager Technology Inc. (Maxager 2002) offers software that computes the profitability of manufacturing a product in real-time. Maxager Precision Bidding™ is designed to communicate with a supplier's ERP and MES systems to get up-to-date data needed to compute profitability. These data can include raw material costs and the production rates on equipment. The profitability is the difference between the customer's price and the current production costs. This is helpful when deciding whether to accept an order, but is not sufficient because it may sometimes be advisable to reject an order with low profit with the expectation that a higher profit order will arrive in the near future. Maxager Precision Bidding aims to address the issue that participants are unwilling to reveal private information by providing suppliers with a system that assists them in responding to RFQs. However, it reduces all the data to profitability, ignoring other important business measures like customer satisfaction, quality, and smooth production line operation.

Yet another aspect of responding to RFQs not addressed by existing art is consideration of other RFQs that the supplier has received in the same time period. A supplier will probably use a different strategy to respond to RFQs when he has received a large number of them for the same product than if the volume of RFQs for that product is small. There is a need for a tool that considers the RFQs received in aggregate when recommending responses.

Gimenez-Funes, in their work use the history of bidding results from previous auctions to estimate the probability of winning an auction with a given bid and situation. They modify bidding strategy based on actual current auction conditions, and find the bid price that maximizes short-term expected benefit of an auction, where the value is determined by the resale price. Finally they choose the decision with the highest global utility. However, the authors do not make use of production scheduling and planning to determine the value of an item, and they do not propose strategies for participating in multiple simultaneous auctions (or RFQs). Furthermore they do not discuss adjusting the bidding strategy based on the bidder's degree of risk.

Conclusions

In this paper, we have presented a system for suggesting how to fulfill a request for quote, taking into account current inventory, production capacity, reworking of goods and purchase of goods from external sources. The solutions provided highlight ways of combining orders from multiple RFQs to improve efficiency and reduce cost. The system provides the human decision maker with the information needed to make good quotes, taking into account possible substitutions and relaxation of requirements that can improve efficiency and profitability.

In future work, we intend to extend the system to suggest quote prices that maximize expected utility and to suggest which attributes of an RFQ a buyer might be willing to compromise on or pay more to have fulfilled.

References

Ariba, IBM, Microsoft; 2002.UDDI Technical White Paper: http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf

Box et al., 2000. Simple Object Access Protocol (SOAP)1.1, W3C Note 08 May 2000 http://www.w3.org/TR/SOA Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, Dave Winer

Clement t, 2002. Web services: Why all the buzz? . CNET News http://www.news.com

Gimenez-Funes et al 1998. Designing Bidding Strategies for Trading Agents in Electronic Auctions. E. Gimenez-Funes, L Godo, J.A. Rodriguez-Aguilar, and P. Garcia-Calves. Proceedings of the Third International Conference on Multi-Agent Systems, Paris, France 3-7 July 1998, IEEE Computer Society

IBM 2002. WebSphere Application Server: http://www-3.ibm.com/software/webservers/appserv/

IBM 2000. WebSphere Commerce Business Edition: http://www-3.ibm.com/software/webservers/commerce/wc_be/

IBM 2001. Web Services Toolkit (WSTK): http://www.alphaworks.ibm.com/tech/webservicestoolkit

Keskinocak et al. 1999. Decision-Support for Managing an Electronic Supply Chain", Pinar Keskinocak, Richard Goodwin, Frederick Wu, Rama Akkiraju, Sesh Murthy, Workshop on Electronic Commerce, 1st Asia Pacific Conference on Intelligent Agent Technology, Hong Kong.

Liu, Goodwin Koening 2001. Risk Averse Auction Planning and its integration into Supply Chain Management Systems. Yaxin Liu, Richard Goodwin, Sven Koenig, AAAI spring symposium on Game Theoretic and Decision Theoretic Agents, 2001.

Maxager Technology Inc. 2002. http://www.maxager.com

Murthy et al 1999. Cooperative Multi-Objective Decision-support for the Paper Industry. Murthy S., Akkiraju R., Goodwin R, Wu F,. Interfaces Volume 29 #5 1999.

Ogbuji, U; 2000. Using WSDL in SOAP applications: An introduction to WSDL for SOAP programmers, Fourthought, Inc. Http://www-106.ibm.com/developerworks/webservices/library/ws-soap/index.html

Perfect.com 2002. http://www.Perfect.com

Rachlin et al 1999. A-Teams: An Agent Architecture for Optimization and Decision-Support. John Rachlin, Richard Goodwin, Sesh Murthy, Rama Akkiraju, Fred Wu, Santhosh Kumaran, Raja Das. Intelligent Agents V, Springer-Verlag 1999.