Predictive Planning for Supply Chain Management

David Pardoe and Peter Stone
Department of Computer Sciences
The University of Texas at Austin
{dpardoe, pstone}@cs.utexas.edu

Abstract

Supply chains are ubiquitous in the manufacturing of many complex products. Traditionally, supply chains have been created through the intricate interactions of human representatives of the various companies involved. However recent advances in planning, scheduling, and autonomous agent technologies have sparked an interest, both in academia and in industry, in automating the process. The Trading Agent Competition Supply Chain Management (TAC SCM) scenario provides a unique testbed for studying and prototyping supply chain management agents by providing a competitive environment in which independently created agents can be tested against each other over the course of many simulations. This paper presents the features of TAC SCM from a planning and scheduling perspective and introduces TacTex-05, the champion agent from the 2005 competition. TacTex-05 takes a predictive approach to its many planning and scheduling decisions by estimating future resource availability and constraints. This paper focuses on these aspects of the agent and isolates their impact with controlled empirical tests.

Introduction

In today's industrial world, supply chains are ubiquitous in the manufacturing of many complex products. Traditionally, supply chains have been created through the intricate interactions of human representatives of the various companies involved. However, recent advances in planning, scheduling, and autonomous agent technologies have sparked an interest, both in academia and in industry, in automating the process (Fox, Chionglo, & Barbuceanu 1993) (Sadeh *et al.* 1999) (Chen *et al.* 1999).

From a planning and scheduling perspective, supply chain management simultaneously requires long-range inventory management, mid-range customer negotiations, and short-term factory scheduling, all of which interact closely.

One barrier to supply chain management research is that it can be difficult to benchmark automated strategies in a live business environment, both due to the proprietary nature of the systems and due to the high cost of errors. The Trading Agent Competition Supply Chain Management (TAC SCM) scenario provides a unique testbed for studying and prototyping supply chain management agents by providing a competitive environment in which independently created agents can be tested against each other over the course of many simulations in an open academic setting. In a TAC SCM game, each agent acts as an independent computer manufacturer in a simulated economy. The agent must procure components such as CPUs and memory; decide what types of computers to manufacture from these components as constrained by its factory resources; bid for sales contracts with customers; and decide which computers to deliver to whom and by when.

One crucial challenge in supply chain management is that decisions must often be made in the face of considerable uncertainty. For instance, purchases of production resources may need to be negotiated long before accurate information about customer preferences becomes available. This challenge is particularly evident in TAC SCM, where sources of uncertainty include the capacity of suppliers to deliver components, the nature of customer demand, and the actions of other agents as they compete for components and customers.

To address this uncertainty, our agent for TAC SCM, TacTex-05, takes a predictive approach to its many planning and scheduling decisions. In particular, TacTex-05 makes predictions concerning the types and quantities of computers that will be requested by customers, the capacities of component suppliers and the prices they are likely to offer, and the probability that an offer to a customer will be accepted at a particular price. Planning and scheduling takes place using these predictions.

The remainder of this paper is organized as follows. First, we summarize the TAC SCM scenario emphasizing its features and challenges from a planning and scheduling perspective. Next, we introduce TacTex-05, the champion agent from the 2005 competition, paying special attention to its predictive approach to its many planning and scheduling decisions. We then isolate the impact of various agent components with controlled empirical tests.

The TAC Supply Chain Management Scenario

In this section, we provide a summary of the TAC SCM scenario. Full details are available in the official specification document[1].

In a TAC SCM game, six agents act as computer manufacturers in a simulated economy that is managed by a game server. The length of a game is 220 simulated days, with each day lasting 15 seconds of real time. At the beginning of each day, agents receive messages from the game server with information concerning the state of the game, such as the customer requests for quotes (RFQs) for that day, and agents have until the end of the day to send messages to the server indicating their actions for that day, such as making offers to customers. The game can be divided into three parts: i) component procurement, ii) computer sales, and iii) production and delivery as expanded on below and illustrated in Figure 1.

Component Procurement

The computers are made from four components: CPUs, motherboards, memory, and hard drives, each of which come

[1] `www.sics.se/tac/tac05scmspec_v157.pdf`

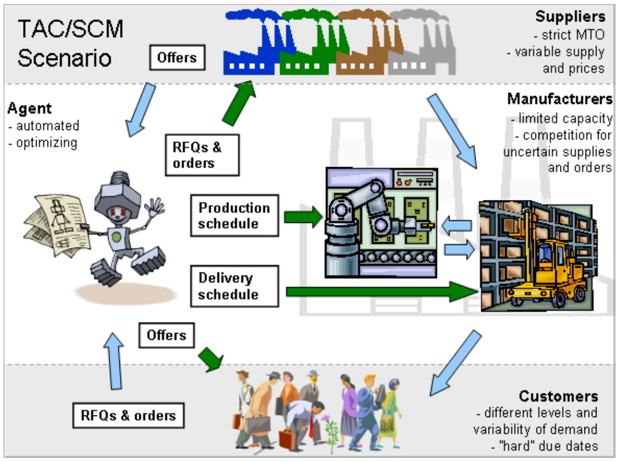

Figure 1: The TAC SCM Scenario (from the official specs[1]).

in multiple varieties. From these components, 16 different computer configurations can be made. Each component has a *base price* that is used as a reference point by suppliers making offers.

Agents wanting to purchase components send requests for quotes (RFQs) to suppliers indicating the *type and quantity* of components desired, the *date* on which they should be delivered, and a *reserve price* stating the maximum amount the agent is willing to pay. Agents are limited to sending at most 5 RFQs per component per supplier per day. Suppliers respond to RFQs the next day by offering a price for the requested components if the request can be satisfied. Agents may then accept or reject the offers.

Suppliers have a *limited capacity* for producing components, and this capacity varies throughout the game according to a random walk. Suppliers base their prices offered in response to RFQs on the fraction of their capacity that is currently free. When determining prices for RFQs for a particular component, a supplier simulates scheduling the production of all components currently ordered plus those components requested in the RFQs as late as possible. From the production schedule, the supplier can determine the remaining free capacity between the current day and any future day. The price offered in response to an RFQ is equal to the base price of the component discounted by an amount proportional to the fraction of the supplier's capacity free before the due date. Agents may send zero-quantity RFQs to serve as price probes. Due to the nature of the supplier pricing model, it is possible for prices to be as low when components are requested at the last minute as when they are requested well in advance. Agents thus face an interesting *tradeoff*: they may either commit to ordering while knowledge of future customer demand is still limited (see below), or wait to order and risk being unable to purchase needed components.

To prevent agents from driving up prices by sending RFQs with no intention of buying, each supplier keeps track of a *reputation* rating for each agent that represents the fraction of offered components that have been accepted by the agent. If this reputation falls below a minimum acceptable purchase ratio (90% for CPU suppliers, and 45% for others), then the prices and availability of components are affected for that agent. Agents must therefore plan component purchases carefully, sending RFQs only when they believe it is likely that they will accept the offers received.

Computer Sales

Customers wishing to buy computers send the agents RFQs consisting of the *type and quantity* of computer desired, the *due date*, a *reserve price* indicating the maximum amount the customer is willing to pay per computer, and a *penalty* that must be paid for each day the delivery is late. Agents respond to the RFQs by bidding in a first-price auction: the agent offering the lowest price on each RFQ wins the order. Agents are unable to see the prices offered by other agents or even the winning prices, but they do receive a report each day indicating the highest and lowest price at which each type of computer sold on the previous day.

Each RFQ is for between 1 and 20 computers, with due dates ranging from 3 to 12 days in the future, and reserve prices ranging from 75% to 125% of the base price of the requested computer type. (The base price of a computer is equal to the sum of the base prices of its parts.)

The number of RFQs sent by customers each day depends on the level of customer demand, which fluctuates throughout the game. Demand is broken into three segments, each containing about one third of the 16 computer types: high, mid, and low range. Each range has its own level of demand. The total number of RFQs per day ranges between roughly 80 and 320, all of which can be bid upon by all six agents. It is *possible for demand levels to change rapidly*, limiting the ability of agents to plan for the future with confidence.

Production and Delivery

Each agent manages a factory where computers are assembled. Factory operation is constrained by both the components in inventory and assembly cycles. Factories are limited to producing roughly 360 computers per day (depending on their types). Each day an agent must send a production schedule and a delivery schedule to the server indicating its actions for the next day. The production schedule specifies how many of each computer will be assembled by the factory, while the delivery schedule indicates which customer orders will be filled from the completed computers in inventory. Agents are required to pay a small daily storage fee for all components in inventory at the factory. This cost is sufficiently high to discourage agents from holding large inventories of components for long periods.

Summary

In summary, the TAC SCM scenario presents many interacting planning and scheduling challenges. For example, an agent's strategy for component procurement necessarily depends on current and predicted supplier prices as well the agent's predicted factory availability and customer demand. Similarly, efficient factory scheduling depends on component availability and projected orders; and the computer sales strategy depends on current and projected customer demand as well as projected factory output and supply availability. TAC SCM is therefore a very valuable testbed domain for real-time (iterative) planning and scheduling under uncertainty.

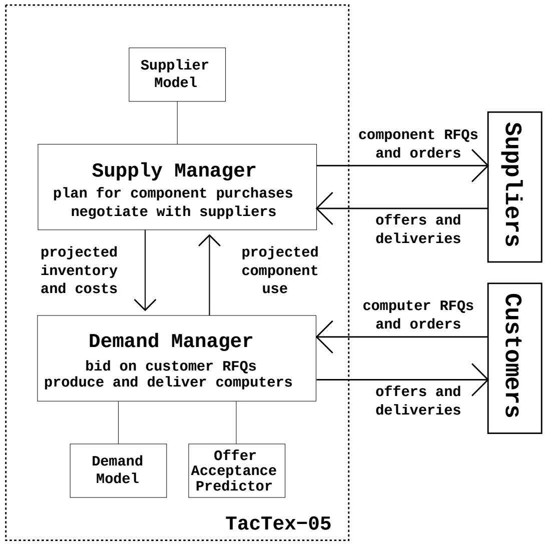

Figure 2: An overview of the main agent components

Overview of TacTex-05

Given the detail and complexity of the TAC SCM scenario, creating an effective agent requires the development of tightly coupled modules for interacting with suppliers, customers, and the factory. The fact that each day's decisions must be made in less than 15 seconds constrains the set of possible approaches.

TacTex-05 is a fully implemented agent that operates within the TAC SCM scenario. We present a high-level overview of the agent in this section, and full details in the sections that follow.

Agent Components

Figure 2 illustrates the basic components of TacTex-05 and their interaction. There are five basic tasks a TAC SCM agent must perform:

1. Sending RFQs to suppliers to request components;
2. Deciding which offers from suppliers to accept;
3. Bidding on RFQs from customers requesting computers;
4. Sending the daily production schedule to the factory;
5. Delivering completed computers.

We assign the first two tasks to a *Supply Manager* module, and the last three to a *Demand Manager* module. The Supply Manager handles all planning related to component inventories and purchases, and requires no information about computer production except for a projection of future component use, which is provided by the Demand Manager. The Demand Manager, in turn, handles all planning related to computer sales and production. The only information about components required by the Demand Manager is a projection of the current inventory and future component deliveries, along with an estimated replacement cost for each component used. This information is provided by the Supply Manager.

We view the tasks to be performed by these two managers as optimization tasks: the Supply Manager tries to minimize the cost of obtaining the components required by the Demand Manager, while the Demand Manager seeks to maximize the profits from computer sales subject to the information provided by the Supply Manager. In order to perform these tasks, the two managers need to be able to make predictions about the results of their actions and the future of the economy. TacTex-05 uses three predictive models to assist the managers with these predictions: a predictive *Supplier Model*, a predictive *Demand Model*, and an *Offer Acceptance Predictor*.

- Record information received from the server and update prediction modules
- The Supply Manager takes the supplier offers as input and performs the following:
  - decide which offers to accept
  - update projected future inventory
  - update replacement costs
- The Demand Manager takes customer RFQs, current orders, projected inventory, and replacement costs as input and performs the following:
  - predict future customer demand using the Demand Model
  - use the Offer Acceptance Predictor to generate acceptance functions for RFQs
  - schedule production several days into the future
  - extract the current day's production, delivery, and bids from the schedule
  - update projected future component use
- The Supply Manager takes the projected future component use as input and performs the following:
  - determine the future deliveries needed to maintain a threshold inventory
  - use the Supplier Model to predict future component prices
  - decide what RFQs need to be sent on the current day

Table 1: Overview of the steps taken each day by TacTex-05.

The Supplier Model keeps track of all information available about each supplier, such as TacTex-05 's outstanding orders and the prices that have been offered in response to RFQs. Using this information, the Supplier Model can assist the Supply Manager by making predictions concerning future component availability and prices.

The Demand Model tracks the customer demand in each of the three market segments, and tries to estimate the underlying demand parameters in each segment. With these estimates, it is possible to predict the number of RFQs that will be received on any future day. The Demand Manager can then use these predictions to plan for future production.

When deciding what bids to make in response to customer RFQs, the Demand Manager needs to be able to estimate the probability of a particular bid being accepted (which depends on the bidding behavior of the other agents). This prediction is handled by the Offer Acceptance Predictor. Based on past bidding results, the Offer Acceptance Predictor produces a function for each RFQ that maps bid prices to the predicted probability of winning the order.

The steps taken each day by TacTex-05 as it performs the five tasks described previously are presented in Table 1.

The Demand Manager

The Demand Manager handles all computation related to computer sales and production. This section describes the Demand Manager, along with the Demand Predictor and the Offer Acceptance Predictor upon which it relies.

Demand Model

When planning for future computer production, the Demand Manager needs to be able to make predictions about future demand in each market segment. For example, if more RFQs are expected for high range than low range computers, the planned production should reflect this fact. The Demand Model is responsible for making these predictions.

In order to explain its operation, further detail is required about the customer demand model. The state of each demand segment (high, mid, and low range computers) is represented by parameters Q_d and τ_d (both of which are internal to the game server). Q_d represents the expected number of RFQs on day d, and τ_d is the trend in demand (increasing or decreasing) on day d. The actual number of RFQs is generated randomly from a Poisson distribution with Q_d as its mean. The next day's demand, Q_{d+1}, is set to $Q_d\tau_d$, and τ_{d+1} is determined from τ_d according to a random walk.

To predict future demand, the Demand Manager estimates the values of Q_d and τ_d for each segment using an approach first used by the agent DeepMaize in 2003 (Kiekintveld *et al.* 2004). Basically, this is a Bayesian approach that involves maintaining a probability distribution over (Q_d, τ_d) pairs for each segment. The number of RFQs received each day from the segment represents information that can be used to update this distribution, and the distribution over (Q_{d+1}, τ_{d+1}) pairs can then be generated based on the game's demand model. By repeating this last step, the expected value of Q_i can be determined for any future day i and used as the number of RFQs predicted on that day. Full details of the approach are available in (Kiekintveld *et al.* 2004). [2]

Offer Acceptance Predictor

In order to bid on customer RFQs, the Demand Manager needs to be able to predict the orders that will result from the offers it makes. A simple method of prediction would be to estimate the winning price for each RFQ, and assume that any bid below this price would result in an order. Alternatively, for each RFQ the probability of winning the order could be estimated as a function of the current bid. This latter approach is the one implemented by the Offer Acceptance Predictor. For each customer RFQ received, the Offer Acceptance Predictor generates a function mapping the possible bid prices to the probability of acceptance. (The function can thus be viewed as a cumulative distribution function.) In (Pardoe & Stone 2004) we explored the possibility of learning to generate this function based on past games. In TacTex-05, however, we use a simpler approach, due to the observation that the range between daily high and low winning prices for a given computer tends to be fairly low when facing competitive agents. This approach involves two components: a linear heuristic for generating a function, and an adaptive means of revising the heuristic's predictions.

The linear heuristic is based on one used by the agent Botticelli in 2003 (Benisch *et al.* 2004) whereby the CDF generated for each RFQ depends only on the type of computer requested and is generated using linear regression on six data points. Specifically, for each of the past five days, the average price bid by TacTex-05 for the given type of computer is determined, along with the fraction of offers accepted on that day. Each pair results in one data point, and the sixth point represents the highest winning price reported for the given type of computer on the previous day along with an acceptance rate of zero. These points are fit using least squares linear regression to generate a linear function that will be used for all RFQs requesting the given computer type.

The linear function is modified using values we call *day factors*, which are designed to measure the effect of the due date on offer acceptance. The due dates for RFQs range from 3 to 12 days in the future, and a separate day factor is learned for each day in this range. Each day factor is set to the ratio of actual orders received to orders expected based on the linear heuristic, for all recent offers made. When an offer is made on an RFQ, the Offer Acceptance Predictor computes the probability of an order by multiplying the initial prediction by the corresponding day factor. The day factors therefore serve both as a means of gauging the impact of due dates on computer prices and as a mechanism for ensuring that the number of orders received is roughly the number expected.

Demand Manager

The Demand Manager is responsible for bidding on customer RFQs, producing computers, and delivering them to customers. All three tasks can be performed using the same production scheduling algorithm. As these tasks compete for the same resources (components, completed computers, and factory cycles), the Demand Manager begins by planning to satisfy existing orders, and then uses the remaining resources in planning for RFQs. The latest possible due date for an RFQ received on the current day is 12 days in the future, meaning the production schedule for the needed computers must be sent within the next 10 days. The Demand Manager thus always plans for the next 10 days of production. Each day, the Demand Manager i) schedules production of existing orders, ii) schedules production of predicted future orders, and then iii) extracts the next day's production and delivery schedule from the result. The production scheduling algorithm, these three steps, and the means of predicting production beyond 10 days are described in the following sections.

Production Scheduling Algorithm The goal of the production scheduler is to to take a set of orders and determine the 10-day production schedule that maximizes profit, subject to the available resources. The resources provided are:

- A fixed number of factory cycles per day;
- The components in inventory;
- The components projected to be delivered; and
- Completed computers in inventory.

The profit for each order is equal to its price (if it could be delivered) minus any penalties for late delivery and the replacement costs for the components involved as specified by the Supply Manager.

The scheduling algorithm used by the Demand Manager is a greedy algorithm that attempts to produce each order as late as possible. Orders are sorted by profit, and the scheduler tries to produce each order using cycles and components from the latest possible dates. If any part of the order cannot be produced, the needed computers will be taken from the existing inventory of completed computers, if possible. The purpose of scheduling production as late as possible is to preserve resources that might be needed by orders with earlier due dates. A record is kept of what production took place on each day and how each order was filled.

[2] The DeepMaize team has released their code for this approach: `www.eecs.umich.edu/~ckiekint/downloads/DeepMaize_CustomerDemand_Release.tar.gz`

It should be noted that the scheduling problem at hand lends itself to the use of linear programming to determine an optimal solution. We initially experimented with this approach, using a linear program similar to one designed for a slightly simplified scenario by (Benisch *et al.* 2004). However, due to the game's time constraints (15s allowed per simulated day), the need to use the scheduler multiple times per day (and in a modified fashion for bidding on customer RFQs), and the observation that the greedy approach is close to optimal, we chose to use the greedy approach.

Handling Existing Orders The Demand Manager plans for the production of existing orders in two steps. Before starting, the production resources are initialized using the values provided by the Supply Manager. Then the production scheduler is applied to the set of orders due in one day or less. All orders that can be taken from inventory (hopefully be all of them to avoid penalties) are scheduled for delivery the next day. The production scheduler is next applied to the remaining orders. No deliveries are scheduled at this time, because there is no reward for early delivery.

Bidding on RFQs and Handling Predicted Orders The goal of the Demand Manager is now to identify the set of bids in response to customer RFQs that will maximize the expected profit from using the remaining production resources for the next 10 days, and to schedule production of the resulting predicted orders. The profit depends not only on the RFQs being bid on on the current day, but also on RFQs that will be received on later days for computers due during the period. If these future RFQs were ignored when selecting the current day's bids, the Demand Manager might plan to use up all available production resources on the current RFQs, leaving it unable to bid on future RFQs. One way to address this issue would be to restrict the resources available to the agent for production of the computers being bid on. Instead, the Demand Manager generates a predicted set of all RFQs, using the levels of customer demand predicted by the Demand Model, that will be received for computers due during the period, and chooses bids for these RFQs at the same time as the actual RFQs from the current day.

Once the predicted RFQs are generated, the Offer Acceptance Predictor is used to generate an acceptance prediction function for every RFQ, both real and predicted. The Demand Manager then considers the production resources remaining, the set of RFQs, and the set of acceptance prediction functions and simultaneously generates a set of bids on RFQs and a production schedule that produces the expected resulting orders. As this process is described fully in (Pardoe & Stone 2004) , we provide only a summary here.

The key to the process is the simplifying assumption that the expected number of computers ordered for each RFQ will be the actual number ordered. In other words, we pretend that it is possible to win a *partial order*, so that instead of winning an entire order with probability p, a fraction p of an order is won with probability 1. With this notion of partial orders, the problem of bid selection is transformed into the problem of finding the most profitable set of partial orders that can be filled with the resources available. This problem can be solved using a variation of the greedy production scheduler described above. Instead of scheduling the production of complete orders, the scheduler considers the production of partial orders resulting from RFQs.

Completing Production and Delivery After applying the production scheduler to the current orders and RFQs, the Demand Manager is left with a 10-day production schedule, a record of how each order was filled, and a set of bids for the actual and predicted RFQs. The bids on actual RFQs can be sent directly to customers in their current form, and computers scheduled for delivery can be shipped. The Demand Manager then considers modifications to the production schedule to send to the factory for the next day. If there are no cycles remaining on the first day of the 10-day production schedule, the first day can be sent unchanged to the factory. Otherwise, the Delivery Manager shifts production from future days into the first day so as to utilize all cycles, if possible.

Production Beyond 10 Days The components purchased by the Supply Manager depend on the component use projected by the Demand Manager. If we want to allow the possibility of ordering components more than 10 days in advance, the Demand Manager must be able to project its component use beyond the 10-day period for which it plans production. One possibility we considered was to extend this period and predict RFQs farther into the future. Another was to predict future computer and component prices by estimating our opponents' inventories and predicting their future behavior. Neither method provided accurate predictions of the future, and both resulted in large swings in projected component use from one day to the next. The Demand Manager thus uses a simple and conservative prediction of future component use.

The Demand Manager attempts to predict its component use for the period between 11 and 40 days in the future. Before 11 days, the components used in the 10-day production schedule are used as the prediction, and situations in which it is advantageous to order components more than 40 days in advance appear to be rare. The Demand Model is used to predict customer demand during this period, and the Demand Manager assumes that it will win, and thus need to produce, some fraction of this demand. While this method of projecting component use yields reasonable results, improving the prediction is a significant area for future work.

The Supply Manager

The Supply Manager is responsible for purchasing components from suppliers based on the projection of future component use provided by the Demand Manager, and for informing the Demand Manager of expected component deliveries and replacement costs. In order to be effective, the Supply Manager must be able to predict future component availability and prices. The Supplier Model assists in these predictions.

Supplier Model

The Supplier Model keeps track of all information sent to and received from suppliers. This information is used to model the state of each supplier, allowing predictions to be made. The Supplier Model performs three main tasks: predicting component prices, tracking reputation, and generating probe RFQs to improve its models.

Price Prediction To assist the Supply Manager in choosing which RFQs to send to suppliers, the Supplier Model predicts the price that a supplier will offer in response to an RFQ with a given quantity and due date. The Supplier Model requires an estimate of each supplier's existing commitments in order to make this prediction.

Recall that the price offered in response to an RFQ requesting delivery on a given day is determined entirely by the fraction of the supplier's capacity that is committed through that day. As a result, the Supplier Model can compute this fraction from the price offered. If two offers with different due dates are available, the fraction of the supplier's capacity that is committed in the period between the first and second date can be determined by subtracting the total capacity committed before the first date from that committed before the second. With enough offers, the Supplier Model can form a reasonable estimate of the fraction of capacity committed by a supplier on any single day.

For each supplier and supply line, the Supply Manager maintains an estimate of free capacity, and updates this estimate daily based on offers received. Using this estimate, the Supplier Model is able to make predictions on the price a supplier will offer for a particular RFQ.

Reputation When deciding which RFQs to send, the Supply Manager needs to be careful to maintain a good reputation with suppliers. Each supplier has a minimum acceptable purchase ratio, and the Supply Manager tries to keep this ratio above the minimum. The Supplier Model tracks the offers accepted from each supplier and informs the Supply Manager of the quantity of offered components that can be rejected from each supplier before the ratio falls below the minimum.

Price Probes The Supply Manager will often not need to use the full five RFQs allowed each day per supplier line. In these cases, the remaining RFQs can be used as zero-quantity price probes to improve the Supplier Model's estimate of a supplier's committed capacity. For each supplier line, the Supplier Model records the last time each future day has been the due date for an offer received. Each day, the Supply Manager informs the Supplier Model of the number of RFQs available per supplier line to be used as probes. The Supplier Model chooses the due dates for these RFQs by finding dates that have been used as due dates least recently.

Supply Manager

The Supply Manager's goal is to obtain the components that the Demand Manager projects it will use at the lowest possible cost. This process is divided into two steps: first the Supply Manager decides *what* components will need to be delivered, and then it decides *how* best to ensure the delivery of these components. These two steps are described below, along with an alternative means of obtaining components.

Deciding What to Order The Supply Manager seeks to keep the inventory of each component above a certain threshold. This threshold is 800, or 400 in the case of CPUs, and decreases linearly to zero between days 195 and 215. Each day the Supply Manager determines the deliveries that will be needed to maintain the threshold on each day in the future. Starting with the current component inventory, the Supply Manager moves through each future day, adding the deliveries from suppliers expected for that day, subtracting the amount projected to be used by the Demand Manager for that day, and making a note of any new deliveries needed to maintain the threshold. The result is a list of needed deliveries that we will call *intended deliveries*. When informing the Demand Manager of the expected future component deliveries, the Supply Manager will add these intended deliveries to the actual deliveries expected from previously placed component orders. The idea is that although the Supply Manager has not yet placed the orders guaranteeing these deliveries, it intends to, and is willing to make a commitment to the Demand Manager to have these components available.

Because prices offered in response to short term RFQs can be very unpredictable, the Supply Manager never makes plans to send RFQs requesting delivery in less than five days. (One exception is discussed later.) As discussed previously, no component use is projected beyond 40 days in the future, meaning that the intended deliveries fall in the period between five and 40 days in the future.

Deciding How to Order Once the Supply Manager has determined the intended deliveries, it must decide how to ensure their delivery at the lowest possible cost. We simplify this task by requiring that for each component and day, that day's intended delivery will be supplied by a single order with that day as the due date. Thus, the only decisions left for the Supply Manager are when to send the RFQ and which supplier to send it to. For each individual intended delivery, the Supply Manager predicts whether sending the RFQ immediately will result in a lower offered price than waiting for some future day, and sends the RFQ if this is the case.

In order to make this prediction correctly, the Supply Manager would need to know the prices that would be offered by a supplier on any future day. Although this information is clearly not available, the Supplier Model does have the ability to predict the prices that would be offered by a supplier for any RFQ sent on the current day. To enable the Supply Manager to extend these predictions into the future, we make the simplifying assumption that the price pattern predicted on the current day will remain the same on all future days. In other words, if an RFQ sent on the current day due in i days would result in a certain price, then sending an RFQ on any future future day d due on day $d + i$ would result in the same price. This assumption is not entirely unrealistic due to the fact that agents tend to order components a certain number of days in advance, and this number generally changes slowly. Essentially, we are saying, "Given the current ordering pattern of other agents, prices are lowest when RFQs are sent x days in advance of the due date, so plan to send all RFQs x days in advance."

The resulting procedure followed by the Supply Manager is as follows. For each intended delivery, the Supplier Model is asked to predict the prices that would result from sending RFQs today with various due dates requesting the needed quantity. A price is predicted for each due date between 5 and 40 days in the future. If there are two suppliers, the lower price is used. If the intended delivery is needed in i days, and the price for ordering i days in advance is lower than that of any smaller number of days, the Supply Manager will send the RFQ. Any spare RFQs will be offered to the Supplier Model to use as probes.

The final step is to predict the replacement cost of each component. The Supply Manager assumes that any need for additional components that results from the decisions of the Demand Manager will be felt on the first day on which components are currently needed, i.e., the day with the first intended delivery. Therefore, for each component's replacement cost, the Supply Manager uses the lowest price found when considering the first intended delivery of that component, even if no RFQ was sent.

For each RFQ, a reserve price somewhat higher than the expected offer price is used. Because the Supply Manager believes that the RFQs it sends are the ones that will result in the lowest possible prices, all offers are accepted. If the reserve price cannot be met, the Supplier Model's predictions will be updated accordingly and the Supply Manager will try again the next day.

3-Day RFQs As mentioned previously, the prices offered in response to RFQs requesting near-immediate delivery are very unpredictable. If the Supply Manager were to wait until the last minute to send RFQs in hopes of low prices, it might frequently end up paying more than expected or be unable to buy the components at all. To allow for the possibility of getting low priced short-term orders without risk, the Supply Manager sends RFQs due in 3 days, the minimum possible, for small quantities in addition to what is required by the intended deliveries. If the prices offered are lower than those expected from the normal RFQs, the offers will be accepted.

The size of each 3-day RFQ depends on the need for components, the reputation with the supplier, and the success of past 3-day RFQs. Because the Supply Manager may reject many of the offers resulting from 3-day RFQs, it is possible for the agent's reputation with a supplier to fall below the acceptable purchase ratio. The Supplier Model determines the maximum amount from each supplier that can be rejected before this happens, and the quantity requested is kept below this amount.

The Supply Manager decides whether to accept an offer resulting from a 3-day RFQ by comparing the price to the replacement cost and the prices in offers resulting from normal RFQs for that component. If the offer's price is lower than any of these other prices, the offer is accepted. If the quantity in another, more expensive offer is smaller than the quantity of the 3-day RFQ, then that offer may safely be rejected.

The 3-day RFQs enable the agent to be opportunistic in taking advantage of short-term bargains on components without being dependent on the availability of such bargains.

2005 Competition Results

Out of 32 teams that initially entered the 2005 TAC SCM competition, 24 advanced past a seeding round to participate in the finals, held over three days at IJCAI 2005. On each day of the finals, half of the teams were eliminated, until six remained for the final day. Game outcomes depended heavily on the six agents competing in each game, as illustrated by the progression of scores over the course of the competition. In the seeding round, TacTex-05 won with an average score of \$14.9 million, and several agents had scores above \$10 million. Making a profit was much more difficult on the final day of competition, however, and TacTex-05 won with an average score of only \$4.7 million. The second highest average score was \$1.6 million, and three agents (each of which averaged at least \$6 million in the seeding round) lost money [3].

Due to the complexity of the TAC SCM scenario and the vast number of decisions that must be made during a single game, it is difficult to isolate the factors that contributed to TacTex-05's success by analyzing game results. When comparing purchases of individual component types or sales of individual computer types on a day-by-day basis[4], it does not appear that TacTex-05 obtained significantly lower purchase prices or significantly higher sales prices than other competitive agents. This fact suggests the possibility that TacTex-05 was better able to focus on the types of computer that were most profitable at any point in time given the component and computer prices then present in the market. Two observations potentially related to this hypothesis are that TacTex-05 tends to carry smaller component inventories throughout the game than many of its competitors, and also appears more flexible in its choice of when to buy components, showing a willingness to purchase components only a short time in advance of their use. An agent that buys components at the last possible moment may be better able to match its purchases to current customer demand. There is the risk, however, that the agent might wait too long and be unable to purchase components at a reasonable price, or at all. This tradeoff, and other possible factors related to TacTex-05's success, are explored in the next section.

Experiments

We now present the results of controlled experiments designed to measure the impact of individual components of TacTex-05 on its overall performance. In each experiment, two versions of TacTex-05 compete: one unaltered agent (which we will call the base agent) that matches the description provided previously, and one agent that has been modified in a specific way. Each experiment includes 30 games. The other four agents competing — Mertacor, MinnieTAC, GoBlueOval, and RationalSCM — are taken from the TAC Agent Repository[5], a collection of agents provided by the teams involved in the competition.

Experimental results are shown in Table 2. Each experiment is labeled with a number. The columns represent the averages over the 30 games of the total score (profit), percent of factory utilization over the game (which is closely correlated with the number of computers sold), revenue from selling computers to customers, component costs, storage costs, penalties for late deliveries, and the percent of the games in which the altered agent outscored the base agent. The final column indicates whether the difference in score observed between the two agents is statistically significant with 99% confidence according to a paired t-test. The first row, experiment 0, is provided to give perspective to the results of other experiments. In experiment 0, two base agents are used, and all numbers represent the actual results obtained. In all other rows, the numbers represent the *differences* between the results of the altered agent and the base agent (from that experiment, not from experiment 0). In general, the results of the

[3] Competition scores are available at `http://www.sics.se/tac/scmserver`

[4] using the CMieux toolkit (Benisch et. al. 2005)

[5] `http://www.sics.se/tac/showagents.php`

base agents are close to those in experiment 0, but there is some variation due to differences between games (e.g. customer demand), and due to the effects of the altered agent on the economy.

We first present experiments designed to measure the importance of prediction accuracy in our predictive planning approach. We then examine the sensitivity of our agent to some of its parameters, particularly those related to the important decision of when to purchase components.

The Three Predictor Modules

The first three sets of experiments probe the sensitivity of the agent to changes in the predictor modules.

Offer Acceptance Recall that for each customer RFQ a function is generated mapping a price offered to the probability of acceptance. This function is generated by multiplying the result of linear regression by a day factor. In experiment 1, no day factors are used, and the score decreases considerably. In experiment 2, day factors are used, but instead of using linear regression to find probabilities across prices, a single price is chosen at which the RFQ is expected to be won. The price chosen is the greater of 95% of the previous day's high price for the computer type, and the previous day's low price. This quantity corresponds roughly to the average selling price of a computer. For any offer below this price, the prediction is made that the offer will be accepted with probability 1, before the day factor is applied. The results show a small, non-significant difference between the use of this heuristic and the use of linear regression, supporting the claim made earlier that the difference between winning prices is small enough to limit the value of learning a detailed acceptance function.

It appears that the use of day factors is the key to the success of the Order Acceptance Predictor. The day factors serve two roles, however: measuring the impact of due date on offer acceptance, and serving as a feedback mechanism to ensure that the number of orders received is in line with the predictions. To measure the relative importance of these two roles, we replace the day factors in experiment 3 with a single multiplier to be used regardless of an RFQ's due date. Like an all-inclusive day factor, the multiplier represents the ratio between all actual and expected orders (i.e. those predicted from the previous days). Linear regression is used as before. The results show the single multiplier to be less effective than the day factors, but much more effective than nothing at all, as in experiment 1. Thus, while considering due dates is of some value in predicting offer acceptance, it appears the feedback role is the more important aspect of the day factors.

Customer Demand In experiment 4, we investigate the value of the Demand Model by ignoring its predictions and instead assuming that the number of RFQs received on any day in the future will be the same as the number received on the current day. Surprisingly, this has an insignificant effect on performance, and we have been unable to determine why.

Component Supply The predictions of prices that will be offered in response to RFQs cannot simply be "turned off" as the predictions were in the previous experiments, because the behavior of the Supply Manager is dependent upon comparisons of prices. Instead, we perform experiments in the following section that modify the behavior of the Supply Manager directly.

Experiments with the Supply Manager

The results to this point highlight the importance of the Offer Acceptance Predictor within TacTex-05 and demonstrate that the Demand Predictor is, at least in the economy under consideration, not important. This section presents experiments to delve deeper into the importance of the Supplier Model, in part by examining TacTex-05's sensitivity to several parameters in the Supply Manager. Ultimately, we discover that the Supplier Model *is* an important part of TacTex-05's overall planning. The experiments further suggest a new strategy for enhancing our baseline agent.

3-day RFQs In experiment 5, 3-day RFQs are not used, preventing the agent from taking advantage of bargains on components requested in the short term. The resulting decrease in score indicates that 3-day RFQs are an important factor in the agent's ability to acquire components at low prices. In addition, the decrease in factory utilization suggests that components purchased from 3-day RFQs do not simply take the place of components that would otherwise have been purchased further in advance. Apparently the prices obtained are sufficiently low to make additional production profitable.

Inventory Threshold The next set of experiments examines the impact of the inventory threshold used. Normally, the Supply Manager attempts to maintain an inventory of at least 800 components of each type beyond the projected use. In experiments 6, 7, 8, 9, and 10, inventory thresholds of 100, 200, 400, 1200, and 1600 are used, respectively. An agent able to perfectly predict future component needs and availability would have no need to maintain surplus inventory, and could plan so that component deliveries would arrive just in time to be used. Maintaining a large inventory can therefore be seen as a way of dealing with inaccuracies in predictions, preventing the lost revenue and penalties that can come from component shortages at the cost of higher storage costs and possibly unnecessary component purchases. This tradeoff is clearly seen in the results. As the inventory threshold increases, factory utilization and revenue increases and penalties decrease, but storage costs increase, and the additional production is not necessarily profitable.

This last fact is somewhat surprising, because it is not immediately clear why simply holding a larger inventory would cause an agent to sell computers at a loss. One possible explanation is that component costs tend to decrease over the course of a game, and therefore an agent that builds up a large surplus inventory early in the game is buying these components at their most expensive, but this cannot account for the entire difference. An analysis of the game logs shows that for the agents using a threshold below the standard 800, the reduced factory utilization is usually caused by a shortage of components, and not a voluntary decision to not produce. Component shortages can only occur when the agent is unable to obtain components that it had planned to buy, indicating that predictions of component availability were too high. In this situation, predicted replacement costs would be too low, possibly causing the agent to sell computers when it is not actually profitable to do so. This idea is supported by the observation that during times when the base agent is

Table 2: Experimental results. In each experiment, one altered version of TacTex-05 and one unaltered version compete in 30 games, along with four additional agents. Columns represent the total score (with standard deviations), percent of factory utilization, revenue from customers, component costs, storage costs, penalties, how often the altered agent outscored the unaltered agent, and an indicator if the difference in score is statistically significant according to a paired t-test with 99% confidence. Numbers represent millions of dollars. In experiment 0, provided to place other experiments' results in perspective, no alteration is made to TacTex-05, and numbers represent the actual results. In all other experiments, numbers represent the difference between the altered and unaltered agent.

Experiment #	*Score*	*Util.*	*Revenue*	*Costs*	*Storage*	*Penalties*	*Win %*	*Significant?*
0	\$2.54M	89%	\$111.25M	\$106.14M	\$1.91M	\$.36M	-	-
1	-2.46 ± 2.65	-1%	-2.25	-5.4	+.70	-.20	0%	**Y**
2	$-.19 \pm .68$	0%	-.08	+.17	+.01	-.08	30%	**N**
3	$-.75 \pm 1.23$	-3%	-4.57	-3.56	+.07	-.30	20%	**Y**
4	$+.05 \pm 1.28$	+1%	+2.45	+2.36	+.01	+.02	40%	**N**
5	-2.36 ± 2.75	-5%	-7.01	-4.18	-.54	+.06	7%	**Y**
6	-3.88 ± 4.74	-44%	-52.24	-47.97	-.81	+.50	7%	**Y**
7	$+.10 \pm 1.65$	-28%	-31.80	-31.41	-.69	+.33	53%	**N**
8	$+1.29 \pm 1.73$	-10%	-12.23	-13.07	-.49	+.17	87%	**Y**
9	-1.66 ± 1.99	+4%	+4.65	+5.84	+.45	-.11	10%	**Y**
10	-2.70 ± 3.02	+4%	+5.89	+7.62	+.86	-.18	3%	**Y**
11	-4.34 ± 4.53	+6%	+6.40	+10.12	+.58	-.20	0%	**Y**
12	-2.13 ± 3.23	-31%	-40.01	-38.55	-.24	+.95	13%	**Y**
13	$-.46 \pm 3.83$	-18%	-25.94	-25.88	-.08	+.51	50%	**N**
14	$-.18 \pm .80$	0%	-.54	-.43	-.03	+.12	40%	**N**
15	$-.20 \pm .83$	+1%	+1.51	+1.70	+.04	-.02	37%	**N**
16	-1.25 ± 1.74	+4%	+5.47	+6.63	+.22	-.19	10%	**Y**
17	$+.03 \pm 3.92$	-15%	-22.42	-22.65	-.04	+.23	70%	**N**
18	$+.78 \pm 1.76$	-9%	-12.26	-13.15	+.03	+.12	73%	**Y**
19	$+.46 \pm .88$	-2%	-3.14	-3.58	-.04	+.01	73%	**Y**
20	$-.41 \pm .89$	+1%	+1.18	+1.51	+.04	+.02	40%	**Y**
21	$+.11 \pm 2.73$	-19%	-24.85	-24.64	-.45	+.26	66%	**N**
22	-6.76 ± 8.00	-48%	-66.96	-60.52	-.38	+.57	0%	**Y**
23	-2.05 ± 2.29	-4%	-5.62	-3.54	-.12	+.06	0%	**Y**

producing computers and the altered agent is not, the base agent's score is often decreasing. The results of this experiment thus suggest that improvements to the component price predictions may be needed in certain situations.

Reduced RFQ Flexibility As mentioned previously, flexibility in the choice of how far in advance to buy components appears to be a distinguishing characteristic of TacTex-05. In experiments 11, 12, and 13, we remove this flexibility and consider three simple alternative methods of deciding when components should be ordered. In experiment 11, no attempt is made to wait for the best day to send RFQs, and RFQs are sent for all needed components immediately. In experiment 12, components are always requested ten days in advance of anticipated need. In experiment 13, components are requested five days in advance. None of these strategies appear effective. In experiment 11, components are purchased in higher quantities and at higher prices than usual, resulting in a very poor score. In experiments 12 and 13, the game logs show that the prices paid for components are not much higher than the prices paid by the base agent. Good prices can apparently be found on these dates much of the time, but not always, as indicated by the decreased factory utilization. Somewhat surprisingly, component availability appears higher when components are requested five days in advance rather than ten, and the relatively small decrease in score in experiment 13 despite the large decrease in factory utilization suggests that the situations in which production is reduced may be those in which it is least profitable, similar to the effect observed with the previous set of experiments.

Range of Days Considered As the choice of when to request components appears important, the next two sets of experiments are designed to measure the effect of restricting the number of days in advance in which the Supply Manager can plan to request components, although to a lesser degree than the previous experiments. Recall that the Supply Manager will normally request components at least five days in advance of anticipated need, and at most 40 days in advance. In experiments 14, 15, and 16, the minimum number of days is changed to four, seven, and ten, respectively, and in experiments 17, 18, 19, and 20, the maximum number of days is changed to 10, 20, 30, and 50, respectively. From experiments 14, 15, and 16, it appears that the best prices can often be found by waiting until fewer than ten days remain. The results of experiments 17 through 20 are somewhat more surprising. It appears that ordering too far in advance, in particular beyond 20 days, is detrimental to performance. This could be due either to incorrectly predicting long term prices, or buying unneeded components due to incorrect projected future component use. Data from the game logs suggests the latter. Prices paid do not seem to differ significantly between the base agent and altered agents. It appears that similarly good prices can be found when ordering either a long or short distance in advance, and that the agents that are restricted from ordering far in advance are able to make purchases that better reflect the current customer demand. Unlike the experiments involving inventory thresholds, it appears that the reduction in factory utilization is planned (not due to component shortages) and occurs during periods of low customer demand, while the base agent is busy using components it purchased in advance before demand decreased.

The success of the changes made in experiments 8 and 18 prompt experiment 21, in which an inventory threshold of 400 and a maximum request distance of 20 days are used. From the result, it appears that these two changes are not complimentary, and that an agent that is unable to request components far in advance may need a larger inventory threshold to offset the risk of being unable to regularly obtain components.

The Effect of Opponent Strategies Based on the previous experiments, it is tempting to conclude that any change that results in TacTex-05 buying components a shorter distance in advance of their use will be beneficial. Even the simple agent in experiment 14 that is required to request all components exactly five days in advance performs nearly as well as the base agent. It is important to consider that the attractiveness of short term purchasing may simply be a feature of the set of agents competing. If all six agents attempted to use such a short term strategy, would it remain effective? To test this, we replaced the four additional agents previously used with four copies of TacTex-05 modified to request components at most ten days in advance. In experiment 22, the agent restricted to requesting all components five days in advance is tested in this environment, and in experiment 23, the agent limited to requesting components at most 20 days in advance is tested. In both cases, the results are much worse than the results against the previously used group of agents. Thus, there must be some value in maintaining the option of long-term component requests.

In light of these experiments, it would appear that the optimal strategy for the Supply Manager is to refrain from requesting components only as long as it appears that the components can still be obtained at reasonable prices in the short term. For some reason, the current strategy results in components being requested before they need to be. One possible solution would be to modify the current strategy so that instead of sending a request as soon as the predicted price is at its lowest point, the request is only sent when it is believed to be unlikely that a reasonably close price can still be obtained. Such a strategy could possibly be implemented by having the Supplier Model predict a distribution over possible prices instead of simply the prices themselves.

Related Work

A number of agent descriptions for TAC SCM have been published presenting various approaches to the tasks faced by an agent. Strategies used for bidding on customer RFQs range from game-theoretic analysis of the economy (Kiekintveld *et al.* 2004) to fuzzy reasoning (He *et al.* 2005). The approach described here, where probabilities of offer acceptance are predicted and used by the factory scheduler, is also used by (Benisch *et al.* 2004). While attention has also been paid to the problem of component procurement, much of it has focused on an unintended feature of the game rules (eliminated in 2005) that caused many agents to purchase the majority of their components at the very beginning of the game (Kiekintveld, Vorobeychik, & Wellman 2005). One exception is (Buffett & Scott 2004), in which the procurement problem is modeled as a Markov Decision Process and dynamic programming is used to identify optimal actions.

Conclusions and Future Work

In this paper, we have presented the features of TAC SCM as a challenge for planning and scheduling research. In addition, we have introduced and analyzed TacTex-05, a champion solution to this problem. TacTex-05 takes a predictive approach to planning in a dynamic, uncertain environment by actively projecting both future environmental conditions and its own future behavior. The experiments presented indicate the importance of various components of TacTex-05 and provide ideas for future development.

As noted earlier, the results of games depend heavily on the agents participating. Since the experiments presented were started, a number of additional agents have been made available through the TAC Agent Repository. Preliminary experiments against various agent combinations have produced qualitatively similar results in most cases.

In this paper, we focus on predictive planning that is based on observations from the current game. In some cases, however, it is also possible to base predictions on data from past games. Indeed, during the final round of the TAC SCM competition, when several games were played against the same set of opponents, TacTex-05 used such adaptation to influence its early and late game strategies. Analysis of TacTex-05's adaptation between games is presented in (Pardoe & Stone 2006), and improving this adaptation is an area of ongoing work.

Acknowledgments

We would like to thank Jan Ulrich and Mark VanMiddlesworth for contributing to the development of TacTex-05, the SICS team for developing the game server, and all teams that have contributed to the agent repository. This research was supported in part by NSF CAREER award IIS-0237699.

References

Benisch, M.; Greenwald, A.; Grypari, I.; Lederman, R.; Naroditskiy, V.; and Tschantz, M. 2004. Botticelli: A supply chain management agent. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 3, 1174–1181.

Benisch et. al. 2005. Cmieux analysis and instrumentation toolkit for TAC SCM. Technical Report CMU-ISRI-05-127, School of Computer Science, Carnegie Mellon University.

Buffett, S., and Scott, N. 2004. An algorithm for procurement in supply chain management. In *AAMAS 2004 Workshop on Trading Agent Design and Analysis*.

Chen, Y.; Peng, Y.; Finin, T.; Labrou, Y.; and Cost, S. 1999. A negotiation-based multi-agent system for supply chain management. *In Workshop on Agent-Based Decision Support in Managing the Internet-Enabled Supply-Chain, at Agents '99*.

Fox, M. S.; Chionglo, J. F.; and Barbuceanu, M. 1993. The integrated supply chain management system. Internal Report, Dept. of Industrial Engineering, University of Toronto, available from `http://www.eil.utoronto.ca/public/iscm-intro.ps`.

He, M.; Rogers, A.; David, E.; and Jennings, N. R. 2005. Designing and evaluating an adaptive trading agent for supply chain management applications. In *IJCAI 2005 Workshop on Trading Agent Design and Analysis*.

Kiekintveld, C.; Wellman, M.; Singh, S.; Estelle, J.; Vorobeychik, Y.; Soni, V.; and Rudary, M. 2004. Distributed feedback control for decision making on supply chains. In *Fourteenth International Conference on Automated Planning and Scheduling*.

Kiekintveld, C.; Vorobeychik, Y.; and Wellman, M. P. 2005. An analysis of the 2004 supply chain management trading agent competition. In *IJCAI 2005 Workshop on Trading Agent Design and Analysis*.

Pardoe, D., and Stone, P. 2004. Bidding for customer orders in TAC SCM. In *AAMAS 2004 Workshop on Agent Mediated Electronic Commerce VI: Theories for and Engineering of Distributed Mechanisms and Systems*.

Pardoe, D., and Stone, P. 2006. TacTex-05: An adaptive agent for TAC SCM. In *AAMAS 2006 Workshop on Trading Agent Design and Analysis / Agent Mediated Electronic Commerce*.

Sadeh, N.; Hildum, D.; Kjenstad, D.; and Tseng, A. 1999. 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*.