The Supply Chain Management Game for the 2006 Trading Agent Competition

John Collins[a], Raghu Arunachalam, Norman Sadeh[b], Joakim Eriksson, Niclas Finne, Sverker Janson[c]

November 2005
CMU-ISRI-05-132

[a]Computer Science Department, University of Minnesota
[b]Institute for Software Research International, Carnegie Mellon University
[c]Swedish Institute of Computer Science, SE-164 29 Kista, Sweden

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This is the specification for the Trading Agent Competition – Supply Chain Management Game for 2006 (TAC SCM-06). Agents are simulations of small manufacturers, who must compete with each other for both supplies and customers, and manage inventories and production facilities.

Based on the positive experience with the 2005 trading agent competition, the changes for 2006 are very minor. They include a configurable ability to learn the identities of competing agents, a slight modification in the handling of agent-supplier reputation, and some clarifications in the text.

The research reported in this paper was funded in part by the National Science Foundation under ITR Grant 0205435.

Keywords: Autonomous Agent, Electronic Commerce, Trading Agent

Contents

1 Background and motivation

Supply chain management is concerned with planning and coordinating the activities of organizations across the supply chain, from raw material procurement to finished goods delivery. In today's global economy, effective supply chain management is vital to the competitiveness of manufacturing enterprises as it directly impacts their ability to meet changing market demands in a timely and cost effective manner. With annual worldwide supply chain transactions in trillions of dollars, the potential impact of performance improvements is tremendous. While today's supply chains are essentially static, relying on long-term relationships among key trading partners, more flexible and dynamic practices offer the prospect of better matches between suppliers and customers as market conditions change. Adoption of such practices has proven elusive, due to the complexity of many supply chain relationships and the difficulty of effectively supporting more dynamic trading practices. TAC SCM was designed to capture many of the challenges involved in supporting dynamic supply chain practices, while keeping the rules of the game simple enough to entice a large number of competitors to submit entries. The game has been designed jointly by a team of researchers from the e-Supply Chain Management Lab at Carnegie Mellon University, the University of Minnesota, and the Swedish Institute of Computer Science (SICS), with input from the research community.

2 Game overview

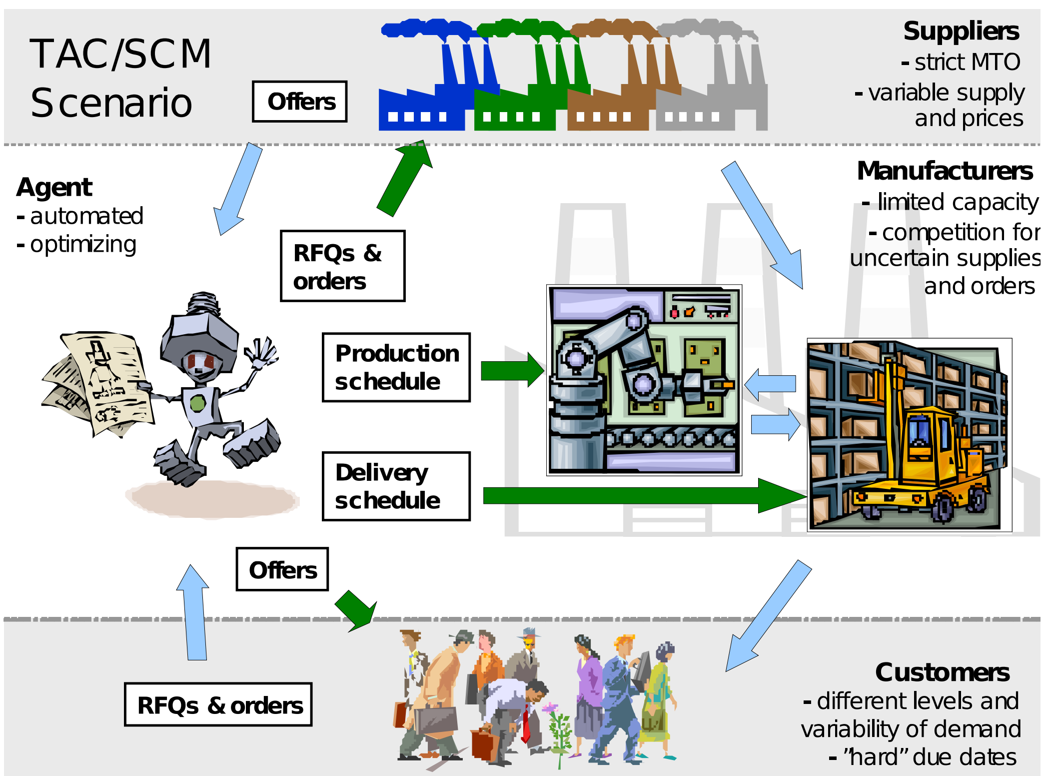

Figure 1: In a TAC SCM game, agents compete for customer orders, manufacture PCs, and procure components.

A TAC SCM game (see Figure 1) consists of a number of “days” or rounds where six personal computer (PC) assembly agents (or “agents” for short) compete for customer orders and for procurement of a variety of components. Each day, customers issue requests for quotes and select from quotes submitted by the agents, based on delivery dates and prices. The agents are limited by the capacity of their assembly lines and have to procure components from a set of eight suppliers. Four types of components are required to build a PC: *CPUs*, *Motherboards*, *Memory*, and *Disk drives*. Each component type is available in multiple versions (e.g. different CPUs, different motherboards, etc.). Customer demand comes in the form of requests for quotes for different types of PCs, each requiring a different combination of components.

A game begins when one or more agents connect to a game server. The server simulates the suppliers and customers, and provides banking, production, and warehousing services to the individual agents. The game continues for a fixed number of simulated days. At the end of a game, the agent with the highest sum of money in the bank is declared the winner.

The game is representative of a broad range of supply chain situations. It is challenging in that it requires agents to concurrently compete in multiple markets (markets for different components on the supply side and markets for different products on the customer side) with interdependencies and incomplete information. It allows agents to strategize (e.g. specializing in particular types of products, stocking up components that are in low supply). To succeed, agents will have to demonstrate their ability to react to variations in customer demand and availability of supplies, as well as adapt to the strategies adopted by other competing agents.

The agents and the various decision processes they must implement are described in Section 3. Section 4 details the behavior of suppliers, and Section 5 describes the behavior of the customers in the game. Section 6 gives details on the components that agents may purchase, and on the products that agents may build and sell. Section 7 gives details on the interaction between the server and the agents, and provides in Table 7 the specific parameters that define a standard tournament game.

3 Agents

Each competitor in a game enters an agent that is responsible for the following tasks:

1. Negotiate supply contracts
2. Bid for customer orders
3. Manage daily assembly activities
4. Ship completed orders to customers

These four tasks are performed autonomously by the agent each day. Human intervention is not allowed during the course of a game. Figure 2 illustrates key daily events involved in running an agent.

At the start of each day, each agent receives:

From Customers:

- Requests For Quotes (RFQs) for PCs.
- Orders won by the agent in response to offers sent to the customers on the previous day.

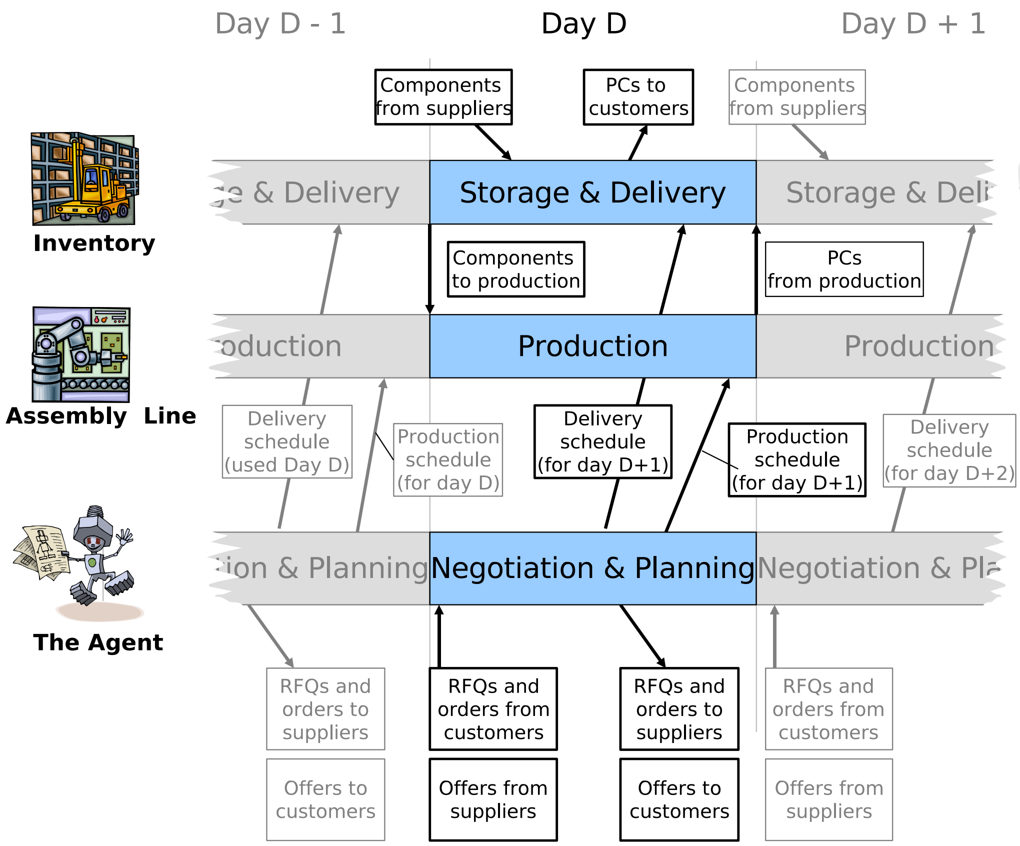

Figure 2: Illustration of a few TAC days, where the agent negotiates with customers and suppliers, and produces and delivers PCs.

- Penalties and order cancellations in response to late deliveries.

From Suppliers:

- Quotes/Offers for components in response to RFQs the agent had sent the day before.
- Delivery of supplies to satisfy earlier orders. The supplies (components) can be used for production the day following delivery. In other words, there is a four-day minimum lag between supplier RFQ and completed products using the parts.

From the Bank:

- Statement of the agent's account.

From the Factory:

- Inventory report, giving quantities of components and finished PCs available.

During the course of a day, each agent must decide:

1. Which customer RFQs to bid on, if any, by returning offers to customers.

2. Which components to (attempt to) procure, by issuing RFQs to suppliers.
3. Which supplier offers to accept, if any, by issuing orders to suppliers.
4. How to allocate available component inventory and factory capacity to the production of PCs, by sending a daily production schedule to the agent's factory.
5. Which assembled PCs to ship to which customers, to satisfy outstanding customer orders, by sending a daily delivery schedule to the agent's factory.

3.1 Production

Each agent is endowed with an identical (simplistic) PC factory containing an assembly cell capable of assembling any type of PC, and a warehouse that stores both components and finished PCs. Each PC type requires a specified number of processing cycles (see the Bill of Materials, Table 5 on page 17), and the agent's assembly cell has a fixed daily capacity.

Each day the agent sends to its factory a production schedule for its assembly cell. The cell will only produce PCs for which the required components are available. PCs in the production schedule are processed sequentially until all capacity has been exhausted. At the end of each day, produced PCs are moved to inventory ready to be shipped the next day.

3.2 Shipping

Shipping is controlled by a delivery schedule, which the agent sends to its factory on a daily basis. The delivery schedule specifies products and quantities to be shipped on the following day, and which customer orders the shipments are to be applied to. All shipments are made from inventory, so that only PCs available in inventory can be shipped. The delivery schedule is processed sequentially until either all shipments have been made or no more shipments can be made due to lack of finished PC inventory. A delivery schedule submitted on day d will cause deliveries to arrive at the customer on day $d+1$. An example (assuming components in inventory):

Day d Before the end of the day, the agent sends a production schedule for production on day $d+1$ to the factory.

Day $d+1$ During the day the factory produces the requested PCs. Before the end of the day the agent sends a delivery schedule, describing the deliveries for day $d+2$, to the factory.

Day $d+2$ PCs are shipped, and will arrive at the customer on the same day.

3.3 Inventory storage costs

Each day, the agent receives a message from its factory giving the quantities of the various components and PCs in inventory. Finished goods and components kept in inventory (components not used for production and PCs not shipped) will be charged a daily storage cost S, which is a percentage of the base price of components. The storage cost will be chosen randomly in the range $[S_{min}, S_{max}]$ (see Table 7 on page 19) at the start of the game and revealed to all the agents. This cost will remain fixed throughout the game and will be applied to the inventory on hand at the end of every day. Storage cost is specified per year (220 days).

3.4 The bank

Each agent has an account in the central bank, and starts the game with no money in the account. Money is added to the account when a customer pays for a product shipment. Money is deducted from the account when agents receive components from suppliers, or when agents default on deliveries and thereby incur penalties. Agents are allowed to go "into the red" or carry a negative balance during the course of the game.

When the agent's balance is negative, the agent is charged interest on a daily basis. The balance is updated daily as

$$b_{d+1} = (1 + \frac{\alpha}{E})b_d + credits_d - debits_d \tag{1}$$

Where b_d is the balance for day d, α is the annual loan interest rate, and E is the length of the game in (simulated) days. A typical annual loan interest rate is $\alpha = 10\%$.

When the agent's balance is positive, the agent is paid a daily interest. This is done by updating the daily balance as

$$b_{d+1} = (1 + \frac{\alpha'}{E})b_d + credits_d - debits_d \tag{2}$$

Typical annual savings interest is $\alpha' = 5\%$.

Values for α and α' are provided to the agent at the beginning of the game (see Table 7 on page 19 for standard tournament values). Every day, the bank notifies each agent of its current bank balance.

4 Suppliers

A standard game includes 8 distinct suppliers. Each component type has two suppliers, both produce all varieties of the component type. The two CPU suppliers specialize in one CPU family: Pintel for Pintel CPUs, IMD for IMD CPUs. Motherboards are supplied by Basus and Macrostar, memories by MEC and Queenmax, and disk drives by Watergate and Mintor. Suppliers are modeled as (approximately) revenue-maximizing entities.

The list of suppliers and their respective products is shown in the component catalog in Table 6 on page 18.

4.1 Supplier model

This section explains how the suppliers in the game are modeled. Each supplier needs to perform three tasks every day:

1. Manage production capacity to satisfy outstanding orders.

2. Make offers to agents based on projected future capacity.

3. Ship components to satisfy outstanding orders.

Suppliers operate under the following assumptions:

1. Suppliers are approximately revenue-maximizing entities.

2. Suppliers engage in a limited form of self-interested risk management by reserving a portion of their future capacity for future business, and by giving preference to agents that have a history of accepting their offers.
3. Suppliers operate on a make-to-order basis.
4. If multiple days of production are required to satisfy an order, inventory is carried over. Inventory carrying costs are assumed to be zero.
5. Only complete orders are shipped, except that on the last day of the game partial orders will be shipped to exhaust inventory.
6. Any excess capacity available on the current day is used to produce components to satisfy future committed orders, if any. However, orders are not shipped before their due dates (or later if not possible to deliver fully on the due date). In the meantime, any inventory built up to satisfy future orders may be diverted to satisfy more short-term (and potentially more profitable) business as long as such diversion leaves enough remaining capacity to meet existing commitments.
7. A random walk is used to determine the production capacity for each day.
8. New sales commitments are made based on expected future capacity.
9. If an existing order cannot be met due to reduced capacity, the order is given priority over orders with a later due date. Thus, delays ripple across the production schedule.
10. Each supplier keeps track of interactions with individual customers (the agents), and gives preference to those customers who have a better record of accepting the supplier's offers.

4.2 Requesting supplies

Each day, each agent may send up to five RFQs to each supplier for each of the products offered by that supplier, for a total of ten RFQs per supplier. Each RFQ r represents a request for a specified quantity q_r of a particular component type c to be delivered on a date $i_r + 1$[1] days in the future, but only if the quoted price is no higher than a reserve price ρ_r per unit. If $\rho_r = 0$, this is interpreted as "unconstrained" and an offer will be generated regardless of unit price. If $q_r = 0$, a price will be quoted for the given due date, but no order can be made. RFQs with due dates beyond the end of the game, or with due dates earlier than 2 days in the future, will not be considered. Once the supplier computes prices, a bid will be generated for each RFQ r, specifying a possibly reduced quantity q'_r, such that the quoted unit price is no higer than ρ_r. If the reserve price constraint cannot be met, even for a single unit, then the quoted quantity q'_r will be 0 (see Section 4.6 for details).

The supplier collects all RFQs received during the day, and processes them together at the end of the day to find a combination of offers that approximately maximizes its revenue. On the following day, the supplier sends back to each agent an offer for each RFQ r, containing the price P_r, adjusted quantity q'_r, and due date. It is possible that the supplier will not be able (or willing – see Section 4.4 below) to supply the entire quantity requested in the RFQ by the due date, even

[1] We use the term i_r to denote the *production lead time* for an RFQ r that has a *delivery due date* of $i_r + 1$.

if the reserve price does not constrain the quantity. In this situation, and only if the reserve price specified in the RFQ is either 0 or high enough[2] – the supplier may respond by issuing up to two amended offers, each of which relaxes one of the two constraints, quantity or due date:

- A *partial* offer is generated if the supplier can deliver only part of the requested quantity on the due date specified in the RFQ (quantity relaxed). Note that a partial offer is distinct from offer quantity reduction due to the reserve-price constraint.

- An *earliest complete offer* is generated to reflect the earliest day (if not past the end of the game) that the supplier can deliver the entire quantity requested (due date relaxed).

Offers are received the day following the submission of RFQs, and the agent must choose whether to accept them. In case an agent attempts to order both the partial offer and the earliest complete offer, the supplier will consider only the order that arrives earlier, and it will ignore the subsequent order for the same RFQ. The example in Section 4.7 illustrates this process in some detail. Note that no offer will be made in case a supplier has no components to offer. This can happen at any time, but is most likely near the end of the game, if all of the supplier's remaining capacity is already committed to agent orders.

Offers made by suppliers are valid only during the day they arrive. If the agent wishes to accept an offer, it must confirm by issuing an order to the supplier.

4.3 Daily production

Every supplier has a dedicated line with some committed and some available capacity, for each component type it supplies. For every component produced by a supplier C^{nom} denotes the nominal capacity. The nominal capacity is the mean capacity and is used by the supplier for planning purposes. The actual production capacity C_d^{ac} for some component on day d is determined by a mean reverting random walk with a lower bound, as follows:

$$C_d^{ac} = \max\left(1, C_{d-1}^{ac} + \text{random}(-0.05, +0.05)C^{nom} + 0.01\left(C^{nom} - C_{d-1}^{ac}\right)\right) \tag{3}$$

where C_{d-1}^{ac} at the start of the game is typically $C^{nom} \pm 35\%$. At the start of every TAC day the supplier computes C_d^{ac} for each of its production lines, and produces components up to that limit, or to satisfy existing orders, whichever is less. In the case of "under capacity" ($C_d^{ac} < C^{nom}$), it is possible that some orders cannot be satisfied by their due dates. Missed orders are queued for delivery on the next possible day, with priority given to the most overdue missed orders. A delivery is made only when the entire quantity of the order can be satisfied. If a delivery can not be made before the game ends, the supplier will deliver as much as it can on the last day.

Tournament values for C^{nom} and for C_{d-1}^{ac} at the start of a game are given in Table 7 on page 19.

4.4 Determining available capacity

A supplier makes offers based on available uncommitted capacity and "divertable" inventory that has been built for future orders. For each of its production lines, a supplier bases its future plans

[2] Where "high enough" is typically somewhere in the range $[0.9 \ldots 1.2]P_{base,c}$ – how high depends on the supplier's current daily capacity, see Section 4.6

on the *expected future capacity* for that line, $C^{ex}_{d,i}$, which gives the expected capacity for each day $d+i$ in the future, based on the actual capacity C^{ac}_d for the current day d.

$$C^{ex}_{d,i} = \begin{cases} C^{ac}_d & : \quad i = 0 \\ 0.99\, C^{ex}_{d,i-1} + 0.01\, C^{nom} & : \quad i > 0 \end{cases} \tag{4}$$

However, suppliers also wish to limit their long-term commitments by reserving some capacity for future business. At any time, each supplier is willing to commit its entire capacity for up to T_{short} days into the future. Beyond that date, the supplier reserves increasing amounts of its daily capacity so that at the beginning of the game, about half of its total capacity is reserved for later orders (at potentially higher prices). If we denote $C^{w}_{d,i}$ as the capacity the supplier is willing to sell for orders due i days in the future, then

$$C^{w}_{d,i} = \begin{cases} C^{ex}_{d,i} & : \quad i \le T_{short} \\ (1 - z\, \max(0, (i - T_{short})))\, C^{ex}_{d,i} & : \quad i > T_{short} \end{cases} \tag{5}$$

Standard game parameters (see Table 7) are $T_{short} = 20$ days and $z = 0.5\%$.

For a day $d+i$, i days in the future from the current day d, $C^{fr}_{d,i}$ denotes the free capacity for that day. $C^{cm}_{d,i}$ denotes the committed capacity for outstanding agent orders that are due to be shipped on day $d+i+1$, since components are shipped no sooner than the day following the day they are produced. Thus,

$$C^{fr}_{d,i} = C^{w}_{d,i} - C^{cm}_{d,i} \tag{6}$$

This means that free capacity on any given day can be positive, if committed order volume due on the following day is less than daily capacity, or it may be negative.

We are now able to determine the capacity that is available to satisfy some set of requests due on day $d+i+1$. That capacity is the sum of the current inventory I, plus the sum of the free capacity between day d and day $d+i$, less any capacity needed to satisfy commitments due after day $d+i+1$.

$$C^{avl}_{d,i} = I + \sum_{j=d+1}^{d+i} C^{fr}_{d,j} + \min_{k \in d+i+1 \ldots E} \left(0, \sum_{j=d+i+1}^{k} C^{fr}_{d,j} \right) \tag{7}$$

Note that current inventory I is computed each day *after* the day's orders have been shipped to agents.

Available capacity is always non-negative (see Equation 16), because shipment dates are adjusted as needed in response to changes in C^{ac}_d.

4.5 Agent reputation

Suppliers give preference in pricing and allocation to agents who are "better" customers (see Section 4.7). To accomplish this, each supplier keeps track of its interaction with each agent a by maintaining an "order ratio" ζ_a for each agent a. This ratio is 1 at the beginning of the game, and it is used to compute a *reputation*, which the supplier uses to discount the expected value of offers made to that agent.

$$\zeta_a = \frac{quantityPurchased_a}{quantityOffered_a} \tag{8}$$

For each supplier, the $quantityOffered_a$ is the sum of the quantities in all the offers issued by the supplier to the agent a thus far, and $quantityPurchased_a$ is the total quantity that agent a

has purchased from the supplier. Note that agents have considerable control over this value, since agents specify a reserve price in their RFQs to the suppliers, and suppliers do not make offers for which computed prices exceed the reserve prices.

If an agent's reserve price for a particular RFQ r cannot be met, then the supplier will compute a reduced quantity q'_r that allows the reserve price constraint to be met, and $quantityOffered_r = q'_r$ (see Section 4.7 for details). If the reserve price for an RFQ can be met, but the quantity cannot be met because of the supplier's capacity limitation, then the agent may receive two offers, a partial offer (see Section 4.7.2) and an earliest-complete offer (see Section 4.7.3), and the agent may order either but not both of them. However, for the purpose of computing the reputation impact of a particular RFQ r, $quantityOffered_r$ will be the greatest of the following three quantities:

1. The quantity in the partial offer,
2. The quantity actually ordered, in case the agent accepts the earliest-complete offer, or
3. 20% of the possibly reduced quantity q'_r.

Since agents should not be punished for requesting the same components from two suppliers and purchasing from only one, we define a factor apr to represent the "acceptable purchase ratio". There are two different values for this parameter. The value apr_1 is used by suppliers of single-sourced components (CPUs), and the value apr_2 is used by suppliers of components that have multiple suppliers (all components other than CPUs). See Table 7 for tournament values.

The reputation rep_a for an agent a is then calculated as

$$rep_a = \frac{\min(apr, \zeta_a)}{apr} \tag{9}$$

This mechanism is used to discourage agents from repeatedly requesting large quantities of a particular component, while having no intention to purchase. The objective of such a strategy could be, for example, to block other genuinely interested agents from procuring supplies, or to raise the prices of supplies for other agents. An agent that employs such a blocking strategy repeatedly would have a low rep_a, and hence each offer made against its RFQs is assumed to produce a lower contribution to the supplier's profit than indicated by the offer price.

Suppliers have a somewhat forgiving nature, and so at the beginning of the game, agents start out with an initial reputation endowment, having a nominal value of $quantityPurchased_0 = quantityOffered_0 = 2000$ (but see Table 7 for tournament values). Also, in order to allow agents to recover from a bad reputation, each agent a given an increment of 100 units each day to both $quantityPurchased_a$ and $quantityOffered_a$. This will allow an agent to recover from turning down a 10,000 unit offer in about 100 days or less.

4.6 Supplier pricing

Pricing is based on the ratio of demand to supply. Higher ratios result in higher prices. When price is computed for an agent with $reputation = y$, demand is the total of the quantities of offers to be made against the current day's RFQs from agents with $reputation \geq y$, plus existing unsatisfied commitments. Supply, for the purpose of price computation, is the current day's production capacity C_d^{ac} projected into the future. It is not the expected capacity C^{ex}. This causes prices to rise and fall with variations in production capacity.

On any day d the offer price of some component c that is due on day $d+i+1$ (which must be produced by day $d+i$) is given by

$$P_{d,i} = P_c^{base}\left(1 - \delta\left(\frac{C_{d,i}^{avl'}}{i\,C_d^{ac}}\right)\right) \tag{10}$$

where

- $P_{d,i}$ is the offer price on day d for an RFQ due on day $d+i+1$
- δ is the price discount factor and has a standard tournament value of 50%
- P_c^{base} is the baseline price for components of type c, given in Table 6
- C_d^{ac} is the supplier's actual capacity on day d as given in Equation 3
- $C_{d,i}^{avl'}$ is similar to $C_{d,i}^{avl}$ given in Equation 7, except that it ignores capacity limits, applies existing inventory I to existing demand only, and assumes that all RFQs for which offers are sent on day d will become committed. For clarity, we split availability into two terms

  $$C_{d,i}^{avl'} = C_{d,i}^{prior} + C_{d,i}^{post} \tag{11}$$

  where $C_{d,i}^{prior}$ is the availability prior to day i

  $$C_{d,i}^{prior} = i\,C_d^{ac} - \sum_{j=d+1}^{i} C_{d,j}^{rfq} + min\left(0,\, I - \sum_{j=d+1}^{i} C_{d,j}^{cm}\right) \tag{12}$$

  and $C_{d,i}^{post}$ is the negative of the capacity required on or before day i to meet commitments that are due on later days. Note that it is always the case that $C_{d,i}^{post} \leq 0$.

  $$C_{d,i}^{post} = \min_{k\in(d+i+1)\ldots E}\left(0,\,(k-d-i)\,C_d^{ac} - \sum_{j=d+i+1}^{k} C_{d,j}^{rfq} + min\left(0,\, I_{d+i}^{post} - \sum_{j=d+i+1}^{k} C_{d,j}^{cm}\right)\right) \tag{13}$$

  where I_{d+i}^{post} is the inventory remaining after computing $C_{d,i}^{prior}$, which is available to apply to commitments with due dates past day $d+i+1$.

  $$I_{d+i}^{post} = \max\left(0,\, I - \sum_{j=d+1}^{i} C_{d,j}^{cm}\right) \tag{14}$$

  In these formulas, $C_{d,j}^{rfq}$ is the total offer quantity due to be shipped on day $d+j+1$ represented by the set of agent RFQs under consideration (for some value y, this is the set of RFQs from agents with $reputation \geq y$).

Note that the computed offer price $P_{d,i}$ can be higher than P_c^{base}, since it is possible that $C_{d,i}^{avl'} < 0$.

4.7 Offer processing

Each supplier generates offers against agent RFQs in a way that approximately maximizes its expected revenue, treats all agents in a fair, predictable way, gives preference to agents with higher reputations (on the assumption that offers to them are more likely to be converted to orders), and reserves some portion of future capacity for future business. As a first approximation, given the set of RFQs $\mathcal{R}_c$ for a component c, the supplier attempts to maximize the objective function

$$\sum_{r \in \mathcal{R}_c} q'_r \, P_{d,i_r} \tag{15}$$

where $q'_r \leq q_r$ is the quantity offered for RFQ r, and i_r is the production lead time for RFQ r. The values of q'_r are limited both by the capacity constraint

$$\forall k \in d+1 \ldots E, \sum_{j=d+1}^{k} (C^{ex}_{d,j} - C^{cm}_{d,j}) \geq 0 \tag{16}$$

and by the reserve-price constraint (see Equation 20). The supplier will make an offer with $q'_r = 0$ if the offer price would be above the agent's reserve price for the RFQ, and it will make an offer with $q'_r < q_r$ if necessary to satisfy either the reserve-price constraint or the capacity constraint.

Given the objective function, the pricing function, and the constraints, the supplier composes a set of offers that maximizes its revenue. But since price is a monotonically increasing function of committed capacity $C^{cm}_{d,i}$ on any given day $d+i$, it is a reasonable approximation to maximize the quantity offered. All that is required is to maximize

$$\sum_{r \in \mathcal{R}_c} q'_r \tag{17}$$

There may be many ways to solve this problem, but there is a reasonably simple approach that is predictable and achieves an intuitive "fairness." We describe it here with an example. In Table 1, we see a few days of capacity and commitments for one of the component assembly lines of a supplier. We assume $d = 16$, $P_{base,c} = 100$, $C^{nom} = 2000$, $C^{ac}_{d,0} = 2100$, $I = 100$, and $T_{short} = 5$. For RFQs issued on day 16, offers will be sent and orders returned on day 17, and the earliest possible delivery date is day 18. In the table, the column labeled $C^{cm}_{d,i}$ represents commitments with due dates on day $d+i+1$.

Table 2 shows a set of RFQs the supplier receives for this product. Note that RFQ #9 is a request for zero quantity. This is a price-probe. The supplier will make an offer, giving the price for the requested due date for an order of zero quantity.

To produce offers, the supplier must perform three operations, in order, as outlined in the following sections.

1. Select the set of winning RFQs and set prices (Section 4.7.1). These are all the RFQs that can be satisfied, given the reserve-price constraint, but disregarding capacity constraints. Prices are set in such a way that agents with higher reputation values are not affected by the demand from agents with lower reputation values.

2. If capacity constraints are violated, then generate partial offers in a way that satisfies capacity constraints and gives preference to agents with higher reputations (Section 4.7.2).

3. If possible, generate earliest complete offers for each of the partial offers generated in step 2 (Section 4.7.3).

Table 1: Existing commitments

$d+i$	$C^{w}_{d,i}$	$C^{cm}_{d,i}$	$C^{avl}_{d,i}$
16	2100	1900	300
17	2099	500	3099
18	2098		4794
19	2097	2500	4794
20	2096		6890
21	2095		8985
22	2084	1300	9769
23	2062	1000	10831
24	2030		12861

Table 2: Agent RFQs

r	rep.	q_r	ρ_r	$d+i_r$
1	1.0	1000	85	19
2	0.9	900	70	21
3	0.7	1500	0	17
4	0.9	500	95	21
5	1.0	200	90	23
6	0.9	2000	95	18
7	0.6	600	90	21
8	0.9	1000	90	17
9	1.0	0	0	20

4.7.1 Select winning RFQs

The set of winning RFQs for a given product is the set that yields the highest expected revenue (and therefore the largest commitment of capacity) that can be satisfied within the reserve price constraint. We do this by sorting the incoming RFQs by decreasing reputation, and evaluating the reserve-price constraint among sets with equal reputation. At this stage we disregard capacity constraints, because we want to base prices on actual demand and not limit prices by the portion of demand the supplier can actually satisfy.

We denote the set of RFQs for a particular product from agents with equal reputation values *reputation* as $\mathcal{R}_{rep}$. Prices are set for each $\mathcal{R}_{rep}$ before considering the next set, which will have a lower value of *reputation*. This prevents agents with lower reputations from affecting prices for agents with higher reputations.

Because the pricing formula in Equation 10 depends on both past and future demand, it is not possible to set prices for RFQs individually. Instead, we must solve a combinatorial problem for each $\mathcal{R}_{rep}$. For each RFQ $r \in \mathcal{R}_{rep}$, this will give us values for both the price P_r and the (possibly reduced) quantity q'_r. Because prices are the same for all orders with the same reputation due on the same day, $P_r = P_{d,i_r}$. Note that an RFQ r for which $q'_r < q_r$ in order to meet its reserve price constraint will receive a partial offer, and if necessary an earliest-complete offer, only for the reduced value q'_r.

To set values for P_r and q'_r, we must maximize the quantity

$$\sum_{r \in \mathcal{R}_{rep}} q'_r \tag{18}$$

subject to the following constraints:

- Offer quantity cannot exceed request quantity.

$$\forall\, r \in \mathcal{R}_{rep},\ q_r \geq q'_r \tag{19}$$

- Offer price cannot exceed reserve price. Given the supplier price formula (Equation 10) we have

$$\forall\, r \in \mathcal{R}_{rep},\ \rho_r \geq P^{base}_c \left(1 - \delta \left(\frac{C^{avl'}_{d,i_r}}{i_r\, C^{ac}_d}\right)\right) \tag{20}$$

into which we substitute Equations 12 and 13. Note that when we substitute Equations 12 and 13, the term $C^{rfq}_{d,j}$ is the *constrained* request quantity of all requests $\mathcal{R}_{rep*,d+j}$ due on day $d+j+1$ from agents with reputations no lower than *rep*. In other words, for some given value of *rep*,

$$C^{rfq}_{d,j} = \sum_{r \in \mathcal{R}_{rep*,d+j}} q'_r \tag{21}$$

- Once the first two constraints are satisfied, remaining ties among RFQs are resolved by giving preference to earlier due dates.

In our example, the RFQs in Table 2 are divided into reputation sets with reputation values of 1.0, 0.9, 0.7, and 0.6, giving us the four sets $\{1,5\}$, $\{2,4,6,8\}$, $\{3\}$, and $\{7\}$. In the first set, RFQs #1 and #5 get prices of 80.3 and 71.7 respectively. In the second set, RFQ #2 does not meet the reserve price constraint, #4 and #6 get prices of 80.6 and 94.7 respectively, and #8 gets a price of 90 at a reduced quantity $q'_8 = 120$. In the third set, RFQ #3 gets a price of 121.3, and in the last set RFQ #7 gets a price of 90 at a reduced quantity $q'_7 = 520$. Table 3 shows the situation after adding RFQs $\{1,3,4,5,6,7,8\}$ to the schedule. The column $C^{avl''}_{d,i}$ shows the available capacity before scheduling the incoming RFQs. All of the reserve price constraints have been satisfied at this point.

Table 3: Final price setting

$d+i$	$C^{avl''}_{d,i}$	RFQs	$C^{fr'}_{d,i}$	$C^{avl'}_{d,i}$	$P_{d,i}$	$q'r$
16	300		200	300		
17	1900	8, 3	-20	-1020	90.0, 121.3	120, 1500
18	3600	6	100	-1020	94.7	2000
19	3600	1	-1400	-1020	80.3	1000
20	5700	9	2100	1080	80.3	0
21	7800	2, 4, 7	1080	2160	80.6, 80.6, 90.0	0, 500, 520
22	8600		800	2960		
23	9700	5	900	3860	71.7	200
24	11800		2100	5960		

Note that the prices quoted for RFQs #4 and #7 and for RFQs #8 and #3 are quite different. This, and the reduction in quantity for RFQ #7, are the results of the differences in reputation between them.

4.7.2 Allocate capacity for partial offers

Once the reserve-price constraint is satisfied and quantities adjusted as necessary to satisfy it, it is possible that capacity constraints will still not be satisfied. We must now adjust offer quantities so that the available capacity (see Equation 7) is non-negative. This constraint is expressed as

$$\forall\, g = [d+1 \ldots E],\, C^{avl}_g \geq 0 \tag{22}$$

In our example, this constraint is violated because the unrestricted reserve price on RFQ #3 allowed for overcommitment. Since current capacity is used for setting prices, while expected capacity is used for making offers, this can result in overcommitment even if computed prices are less than 100% of base. Table 4 shows the capacity situation in our example. We do not have sufficient capacity to satisfy the requests for the first 4 days.

Table 4: Allocation data

$d+i$	$C^{w}_{d,i}$	RFQs	$C^{cm}_{d,i}$	$C^{fr}_{d,i}$	$C^{avl}_{d,i}$	q^{p}_{r}
16	2100		1900	200	300	
17	2099	8, 3	2120	-21	-1026	100, 958
18	2098	6	2000	98	-1026	1660
19	2097	1	3500	-1403	-1026	876
20	2096	9	0	2096	1070	0
21	2095	2, 4, 7	1020	1075	2145	0, 500, 520
22	2084		1300	784	2929	
23	2062	5	1200	862	3791	200
24	2030		0	2030	5821	

In this example, RFQs 8, 3, 6, and 1 form a *conflict set* $\mathcal{R}_{conflict}$, which has an excess demand $C^{conflict}$ of 1026 units.[3] The quantities for partial offers $q^p_r \leq q'_r$ are generated in two steps, according to the agent reputations, as follows:

1. Offer quantities for individual requests cannot exceed actual capacity (we don't make offers for quantities that are physically impossible to produce). Therefore we reduce such individually unrealistic quantities before considering conflicts among requests.

$$\forall\, r \in \mathcal{R}_{rep},\ q'_r \leq \sum_{j=d+1}^{i_r} C^{fr}_{d,j} \tag{23}$$

2. Offer quantities are further reduced, if necessary, to meet the constraint of Equation 22.

$$\forall r \in \mathcal{R}_{conflict},\ q^p_r = q'_r - C^{conflict} \frac{q'_r(1/reputation^m_r)}{\sum_{r' \in \mathcal{R}_{conflict}} q'_{r'}(1/reputation^m_{r'})} \tag{24}$$

where the exponent m controls the relative allocation according to reputation. This formula divides the shortage among the conflicting RFQs in proportion to the respective values of $1/reputation^m_r$. If $m = 3.0$ (the tournament value), then an agent with $reputation = 1.0$ has an advantage of about 37% over an agent with $reputation = 0.9$. The final column in Table 4 shows the adjusted quantities for the 4 RFQs in the conflict set. Note that conflict sets are always identifiable by negative availability on day $d + 1$.

4.7.3 Allocate capacity for earliest-complete offers

The final step in allocation is to generate earliest-complete offers for the RFQs in $\mathcal{R}_{conflict}$. This is done by sorting the RFQs by reputation, highest reputation first. For each RFQ in the sorted

[3] The excess demand in Table 4, represented by the value of $C^{avl}_{d,i}$, is smaller than the "excess" represented by $C^{avl'}_{d,i}$ in Table 3 because the former is based on $C^{w}_{d,i}$ (Equation 5), while the latter is based on $C^{ac}_{d,i}$ (Equation 3).

set, remaining available capacity is allocated to complete the requested quantity, and an earliest-complete offer is generated. In the case of ties in reputation score, the tied requests are allocated equal quantities for each day for which there is non-zero available capacity, as long as unsatisfied requests remain. This will typically give earlier dates to smaller requests.

In our example, RFQ #1 received a complete offer due for shipment on day 20 because of its high reputation, even though it was part of the conflict set. RFQs #6 and #8 are tied with a reputation of 0.9, and both can be delivered on day 21. As it turns out, there is also enough capacity remaining on day 21 to deliver the remainder of RFQ #3. So three earliest-complete offers are made for full delivery of RFQs #3, #6, and #8 on day 21. An earliest-complete offer is not made for RFQ #7, since its partial offer was a result of the reserve-price constraint and not a capacity constraint. The unit price for an earliest-complete offer is the same as the unit price for the corresponding partial offer. A zero-quantity offer will also be made for RFQ #2 with a price of 80.6, which is the price computed for a zero-quantity order to be shipped on day 22 when the requests with a reputation of 0.9 were processed.

4.8 Payment

Agents confirm supplier offers by issuing orders, as outlined above in Section 4.2. Suppliers, wishing perhaps to protect themselves from defaults, will bill agents immediately for a portion *down* of the cost of each order placed. The remainder of the value of the order will be billed when the order is shipped. If a supplier's capacity is below nominal and it hits the end of the game with late, unshipped orders, it will ship what it can on the last day and bill the respective agents. The down payment ratio is typically 10%.

Suppliers ship complete orders to agents on, or as soon as possible after, the due date specified in the associated offer, but not before all orders with earlier due dates have been shipped to their respective agents. The remainder of the offer price is transfered out of the agent's bank account on the day of the shipment. The shipment will show up in the agent's inventory of parts on the same day as the shipment.

5 Customers

Customers request PCs of different types to be delivered by a certain DueDate. Each request is for a quantity chosen uniformly in $[q_{min}, q_{max}]$ (see Table 7). Agents must bid to satisfy the entire order (both quantity and due date) for the customers to regard the bid.

5.1 Customer demand

Customer demand is expressed as requests for quotes (customer RFQs or cRFQs), each of which specifies a product type, quantity q, due date, reserve price ρ, and penalty amount x. For each customer RFQ, the product type and is randomly selected from the available types (see Bill of Materials in Table 5), q is chosen uniformly in the interval $[q_{min}, q_{max}]$, and the due date is the current date plus a uniformly chosen order lead time in the interval $[due_{min}, due_{max}]$. Each customer RFQ also specifies the maximum price per unit that a customer is willing to pay. This reserve price ρ is randomly chosen in the interval $[\rho_{min}, \rho_{max}]$, as shown in Table 7. Customers will not consider any bids with prices greater than ρ. For each customer RFQ, a penalty for late delivery is chosen uniformly in the interval $[\Psi_{min}, \Psi_{max}]$.

Customer requests are classified into three market segments: High range, Mid range, and Low range. For each of these segments, at the start of each day d, customers exhibit their demand by issuing N customer RFQs, according to the following distribution:

$$N = \text{poisson}(Q_d) \tag{25}$$

where Q_d is the "target average" number of customer RFQs for day d issued in each market segment. Q_d will be varied using a trend τ that is updated by a random walk:

$$Q_{d+1} = \min(Q_{max}, \max(Q_{min}, \tau_d Q_d)) \tag{26}$$

$$\tau_{d+1} = \max(\tau_{min}, \min(\tau_{max}, \tau_d + \text{random}(-0.01, 0.01)) \tag{27}$$

Q_0, the start value of Q, is chosen uniformly in the interval $[Q_{min}, Q_{max}]$ (see Table 7), and τ_0, the start value of the τ, is 1.0. The trend τ is reset to 1.0 when the random walk exceeds the minimum or maximum boundaries. In other words, if $\tau_d Q_d < Q_{min}$ or $\tau_d Q_d > Q_{max}$ then $tau_{d+1} = 1.0$. This reduces the bimodal tendency of the random walk.

5.2 Customer bid processing

All agents receive each of the cuatomer RFQs that are generated each day. If the agent wishes to respond to a particular RFQ, it returns a bid to the customer containing a price, a quantity, and a due date.

The customer only considers bids that satisfy all three of the following requirements:

1. the bid promises the entire quantity specified in the RFQ,
2. the bid promises to deliver on the due date specified in the RFQ, and
3. the bid price is below or equal to the reserve price specified by the customer in the RFQ.

All other agent bids are rejected, silently. For each RFQ, the customer collects all the bids that pass the consideration criteria, and selects the bid with the lowest price as the winning bid by issuing an order back to the agent. In case of a tie between two agent bids, the winner is chosen randomly between the tied bids. The winning agent will be notified at the start of the next day by receipt of an order. Also, all agents will be informed daily of the minimum and maximum order prices (P_{min}, P_{max}) for each type of PC ordered the previous day.

5.3 Fulfilling customer orders

Orders are fulfilled when agents ship products to customers (see Section 3.2). Because products shipped on day d arrive on day $d + 1$, shipment must be made on the day preceeding the due date to avoid penalty. Payment is made to the agent's bank account either on the due date, or the day following the shipment date, whichever is later.

Penalties are charged daily when an agent defaults on a promised delivery date, and are automatically withdrawn from the agent's bank account. Penalties are accrued over a period of five days, and after the fifth day the order is cancelled and no further penalties are charged. After the last day of the game all pending orders are charged the remaining penalty (up to five days) since they can never be delivered.

Agents are informed daily of late-delivery penalties and cancellations that result from failure to ship orders to customers as promised.

6 Products and components

The products to be manufactured are personal computers (PCs). Each PC model is built from four component types: CPUs, motherboards, memories, and hard drives.

CPUs and motherboards are available in two different product families, Pintel and IMD. A Pintel CPU only works with a Pintel motherboard while an IMD CPU can be used only with an IMD motherboard. CPUs are available in two speeds, 2.0 and 5.0 GHz, memories in sizes 1 GB and 2 GB, and disks in sizes 300 GB and 500 GB. There are a total of 10 different components, which can be combined into 16 different PC configurations, all of which are described in a Bill of Materials given in Table 5.

Table 5: Bill of Materials

SKU	Components	Cycles	Market segment
1	100, 200, 300, 400	4	Low range
2	100, 200, 300, 401	5	Low range
3	100, 200, 301, 400	5	Mid range
4	100, 200, 301, 401	6	Mid range
5	101, 200, 300, 400	5	Mid range
6	101, 200, 300, 401	6	High range
7	101, 200, 301, 400	6	High range
8	101, 200, 301, 401	7	High range
9	110, 210, 300, 400	4	Low range
10	110, 210, 300, 401	5	Low range
11	110, 210, 301, 400	5	Low range
12	110, 210, 301, 401	6	Mid range
13	111, 210, 300, 400	5	Mid range
14	111, 210, 300, 401	6	Mid range
15	111, 210, 301, 400	6	High range
16	111, 210, 301, 401	7	High range

Each PC type is identified by an integer identifier called a Stock Keeping Unit (SKU). The bill of materials specifies, for each PC type, the constituent components, the number of *assembly cycles* required, and the *market segment* it belongs to. The sixteen PC types are classified into three market segments: High range, Mid range, and Low range. As we have seen, customer demand is expressed independently in each segment.

Table 6 gives the component catalog for a typical game, with information about each component, their base prices, and the suppliers that produce them.

The base price of each component is used to compute the price at which the components are sold (described in Section 4.6) and the range of the customer *reserve price* (described in Section 5.1).

Both the BOM and the component catalog are sent to all agents at the start of the game.

Table 6: Component Catalog

Component	Base price	Supplier	Description
100	1000	Pintel	Pintel CPU, 2.0 GHz
101	1500	Pintel	Pintel CPU, 5.0 GHz
110	1000	IMD	IMD CPU, 2.0 GHz
111	1500	IMD	IMD CPU, 5.0 GHz
200	250	Basus, Macrostar	Pintel motherboard
210	250	Basus, Macrostar	IMD motherboard
300	100	MEC, Queenmax	Memory, 1 GB
301	200	MEC, Queenmax	Memory, 2 GB
400	300	Watergate, Mintor	Hard disk, 300 GB
401	400	Watergate, Mintor	Hard disk, 500 GB

7 Game interaction

Six agents compete in each game. The game takes place over 220 TAC days, each day being 15 seconds long. The agent with the highest sum of money in the bank at the end of the game is declared the winner. The format and content of the various messages exchanged between the agents and the game server are available in the software documentation.

In addition to the interaction with the suppliers and customers, agents (and game viewers) have access to other data within a game.

7.1 Game initialization

At the beginning of each game, each agent receives four initialization messages from the server, as follows:

1. Specific values for variable game parameters (see Table 7 below), including the number of days E in the game, the length in seconds of each day, the supplier acceptable purchase ratio apr, the bank interest rate for debt α and for deposits α', and the annual inventory holding cost S.
2. The Bill of Materials (see Table 5).
3. The Component Catalog (see Table 6).
4. The identities of the agents who have joined the game.

Table 7 gives the game parameters settings for standard competition games.

Values for most of these parameters are sent to the agent at the start of every game. For details see the software documentation.

7.2 Periodic reports of market state

In addition to information that can be gained from the supplier offers and customer demand, periodic reports are generated by the system summarizing the supplier and customer markets.

Table 7: Parameters used in the TAC SCM game

Parameter	Symbol	Standard Game Setting
Length of game	E	220 days
Agent assembly cell capacity		2000 cycles / day
Nominal capacity of supplier assembly lines	C^{nom}	550 components / day
Start capacity of the suppliers assembly lines	C^{ac}_{-1}	$C^{nom} \pm 35\%$
Supplier price discount factor	δ	0.5
Down payment due on placement of supplier order		10%
Acceptable purchase ratio for single-source suppliers	apr_1	0.75
Acceptable purchase ratio for two-source suppliers	apr_2	0.45
Initial reputation endownment		2000
Reputation recovery rate		100 units/day
Average number of customer RFQs in the High and Low range markets	$[Q_{min}, Q_{max}]$	25 – 100 per day
Average number of customer RFQs in the Mid range market	$[Q_{min}, Q_{max}]$	30 – 120 per day
Interval between Market Reports		20 days
RFQ volume trend for customers (all market segments)	$[\tau_{min}, \tau_{max}]$	$[0.95, 1/0.95]$
Range of quantities for individual customer RFQs	$[q_{min}, q_{max}]$	$[1, 20]$
Range of lead time (due date) for customer RFQs		3 to 12 days from the day the RFQ is received
Range of penalties for customer RFQs	$[\Psi_{min}, \Psi_{max}]$	5% to 15% of the customer reserve price per day
Customer Reserve Price		75 – 125% of nominal price of the PC components
Annual bank debt interest rate	$[\alpha_{min}, \alpha_{max}]$	6.0 – 12.0%
Annual bank deposit interest rate	$[\alpha'_{min}, \alpha'_{max}]$	0.5α
Annual storage cost rate	$[S_{min}, S_{max}]$	25% – 50% of nominal price of components
Short-term horizon for supplier commitments	T_{short}	20 days
Daily reduction in supplier available capacity for long-term commitments	z	0.5%
Allocation reduction exponent	m	3.0

Four component type supply reports (one report for CPU, memory, hard disk, and motherboard respectively) are made available every 20 TAC days to all competing agents. The component type supply reports contain the following information for each component:

- Aggregate quantities shipped by all suppliers in the given period.
- Aggregate quantities ordered from all suppliers in the given period.
- Mean price per SKU for all components ordered during the period (this data is specified by SKU, so the price is available for individual CPU, motherboard, memory, and disk models).

For each supplier, the market report gives the mean production capacity C^{ac} for the period. Customer demand data includes request volume, order volume, and the average order price for each PC type during the reporting period.

The market reports allow agents to learn something about the state of the suppliers and customers and, also strategies employed by other competing agents. For instance an agent can detect the lack of availability of a particular PC type in the market from the customer demand reports and choose to target the niche. Similarly supply reports may give insight into supply procuring practices of other agents.

7.3 Other metrics

In order for viewers to follow the game and allow data for post mortem sessions a set of metrics (including the following) will be monitored throughout the game. They can be viewed during a game using a game viewer, but not all of them are accessible to agents.

- Bank balance
- Inventory quantities and cost of inventory held
- Delivery performance
- Assembly cell utilization

References

[1] R. Arunachalam, N. Sadeh, J. Eriksson, N. Finne, and S. Janson. Design of the supply chain trading competition. In *IJCAI-03 Workshop on Trading Agent Design and Analysis*, Acapulco, Mexico, August 2003.

[2] R. Arunachalam, N. Sadeh, J. Eriksson, N. Finne, and S. Janson. The TAC supply chain management game. Technical Report CMU-CS-03-184, Carnegie-Mellon University, 2003.

[3] N. Sadeh, J. Eriksson, N. Finne, and S. Janson. TAC-03: A supply-chain trading competition. *AI Magazine*, 24(1):92–94, 2003.

A Changes from the 2005 TAC-SCM game

There are two small changes between the 2005 and 2006 games, and the intent is that agents that operated correctly under the 2005 rules will continue to operate correctly under the 2006 rules. The changes are:

- Section 4.5: There are now two different values for the "acceptable purchase ratio" for computing reputation. Specifically, the ratio for single-source suppliers is relaxed slightly.
- Section 7.1: Agents may learn the identities of competing agents at the beginning of a game. This is intended to allow agents to make use of opponent modeling.

$Revision: 1.6 $
$Date: 2005/11/08 14:42:21 $