The Supply Chain Management Game for the Trading Agent Competition 2004

Raghu Arunachalam[1], Norman Sadeh[1] , Joakim Eriksson[2], Niclas Finne[2]
and Sverker Janson[2]

CMU-CS-04-107
February 2004

[1]Institute for Software Research International
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891

[2] Swedish Institute of Computer Science
SE-164 29 Kista
Sweden

Also appears as ISRI Technical Report CMU-ISRI-04-104

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

Abstract

This report is the specification for the Trading Agent Competition Supply Chain Management Game - TAC SCM-04, to be held between July 20-22, 2004, in New York in conjunction with AAMAS-04. Based on the experience of the 2003 trading agent competition a few enhancements have been added to the original game. The price function has been modified to better reflect demand; 2) storage costs have been introduced; and 3) customer demand has been segmented into multiple markets.

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 however proven elusive, due to the complexity of many supply chain relationships and the difficulty in 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 and the Swedish Institute of Computer Science (SICS).

2 Game overview

Specifically, TAC SCM features rounds where six personal computer (PC) assembly agents (or "agents" for short) compete for customer orders and for procurement of a variety of components over a period of several months. 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 represented in the game: *CPUs, Motherboards, Memory, and Hard drives.* It features a variety of components of each type (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.

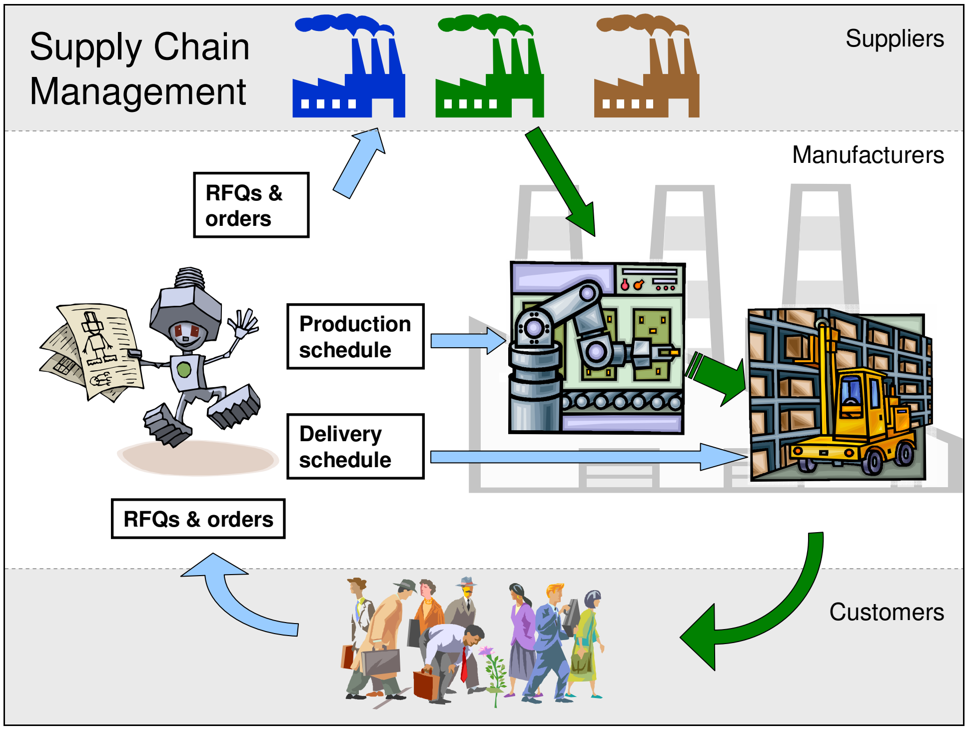

Figure 1: In TAC SCM an agent's task is to manufacture PC's, win customer orders, and procure components.

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. At the end, the agent with the highest sum of money in the bank is declared the winner.

3 Agents

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

1) Negotiate supply contracts

2) Bid for customer orders

3) Manage daily assembly activities

The three tasks are performed by the agent daily. Figure 2 illustrates key daily events involved in running an agent.

A TAC Day

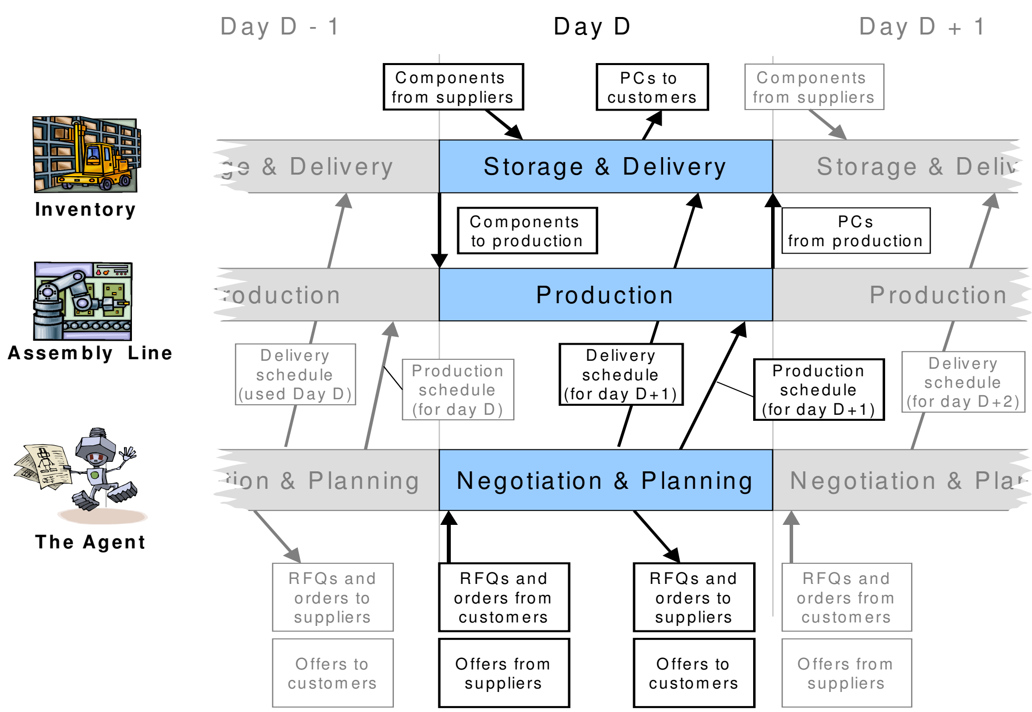

Figure 2: Illustration of a few TAC days, where the agent plans, produces and delivers PCs.

At the start of each day, each agent receives:

From Customers:

Request For Quotes (RFQs) for PCs

Orders won by the agent in response to offers sent to the customer

From Suppliers:

Quotes/Offers for components in response to RFQs the agent had sent them the day before

Delivery of supplies it had ordered earlier. The supplies (components) can be used for production the day after the delivery (i.e. there is a two day lag between request and earliest delivery)

From Bank:

Statement of the account

From Factory

Inventory report (quantity of components and finished PCs available)

During the course of the day, the agent plans:

1) Which customer RFQs to bid on, if at all
2) Which components to procure (and issues RFQs to suppliers for the procurement)
3) Which supplier offers to accept, if any
4) Which PC orders to assemble for, subject to availability of supplies – the list of PCs to be assembled on a given day is referred to as the *production schedule*
5) Which assembled PCs to ship to which customers – the list of PCs to be shipped to customers on a given day is referred to as the *delivery schedule*

Each agent is endowed with an identical (simplistic) PC factory containing an assembly cell capable of assembling any type of PC, and an inventory storing both components and finished PCs. Each PC type requires a pre-specified amount of processing (specified in the next section under Bill of Materials) and each agent has a fixed amount of processing time available each day – its *assembly cell capacity*.

Each day the agent makes a *production schedule* for its assembly cell. The cell only produces the 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.

Shipping is controlled by the delivery schedule. All shipments are made from inventory and only PCs available in inventory can be shipped. The shipping schedule is processed sequentially until either all shipments have been made or no more shipments can be made due to lack of PCs. *Delivery schedule* submitted today, will cause deliveries to the customer the next day. An example (assuming components in inventory):

Day D Before the end of 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 During the day the PCs arrive to the customer

4 Products and components

The products to be manufactured are personal computers (PCs) 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 incorporated only in 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.

4.1 Bill of Materials, Component Catalog

All PCs that can be assembled are described in a Bill of Materials, BOM. The table in figure 3 shows the BOM for TAC SCM. Each PC type (SKU, Stock Keeping Unit) are listed with their constituent components, the number of *assembly cycles* required and the market segment they belong to. The sixteen PCs are classified into three market segments: High range, Mid range, and Low range.

In addition to the BOM there is a component catalog with information about the components, and the suppliers that produce them (see right side of the table below).

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

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	110,210,301,401	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

Component	Base Price	Supplier	Name
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

Figure 3: Bill of Materials and Component Catalog

The base price of each component is used to compute the price at which the components are sold (described in section 5.5) and the range of the customer *reserve price* (described in section 6).

5 Suppliers

Each component type has two suppliers, both produce all varieties of the component type. There are eight suppliers in total. The two CPU suppliers specialize in one CPU family. The suppliers are Pintel for Pintel CPUs, IMD for IMD CPUs, Basus and Macrostar for motherboards, MEC and Queenmax for memories, and Watergate and Mintor for disks.

This information is shown in the component catalog above.

5.1 Requesting supplies

Each day an agent is allowed to send a maximum of ten RFQs to each supplier. This allows an agent to probe the supplier while not swamping it with too many messages. The RFQs are bundled together in an *ordered list*, with the highest priority RFQ at the start (*RFQBundle)*. The supplier collects all RFQs received during the day, and at the end of the day processes them. Processing involves the supplier selecting an RFQ bundle and considering the next unprocessed RFQ in that bundle. This step (selection of a RFQ bundle and processing of the next RFQ) continues until all the RFQs have been processed. Selection of an RFQ bundle by a supplier depends on the *Likelihood* (described below) of the agent that sent the RFQ bundle- higher the *Likelihood* of an agent the more likely (or probable) its RFQ bundle will be the next selected bundle.

Each agent is associated with an order ratio for each supplier. This ratio is 1 in the beginning of the game, and it is used to bias the sequence in which the incoming RFQs are processed.

$$\text{Order ratio} = \frac{QuantityPurchased}{QuantityRequested} \tag{1}$$

For each supplier the *QuantityRequested* is the sum of the quantities in all the RFQs issued by an agent thus far. *QuantityPurchased* is the total quantity that an agent has purchased from the supplier.

Since agents should not be punished for requesting the same components from both suppliers of a specific component the weight is calculated as

$$weight = min(0.5, \text{order ratio}) \tag{2}$$

The likelihood for a specific agent, A, to be selected is:

$$Likelihood_A = \frac{Weight_A}{\sum_{agents} Weight} \tag{3}$$

This mechanism is used to discourage agents employing a strategy that entails repeatedly requesting a large quantity of a particular component, but having no intention of purchase. The objective of such a strategy being to block other genuinely interested agents from procuring supplies. An agent that employs such a blocking strategy repeatedly, would have a low *Likelihood,* and hence a less chance of its RFQs being selected ahead of an agent with a higher *Likelihood,* preventing it from blocking.

If the supplier chooses to respond to the RFQ, a response in the form of an offer is generated. This cycle continues until all RFQs have been considered. As in the case of *RFQBundles* all *offers* destined for an agent are bundled together into *OfferBundles*.

The content of an RFQ is as follows:

> *RFQ ::= <RFQ-Id, Component-type, Quantity, DueDate>*
>
> *RFQ-Id, Component-type, Quantity, DueDate ::= Integer*

Nb. An RFQ with *DueDate* beyond the end of the game will not considered by the supplier.

The supplier responds to the agent with an offer containing:

> *Offer::= <Offer-Id, RFQ-Id, Quantity, DueDate,Price >*
>
> *Offer-Id, RFQ-Id, Quantity,Price ::= Integer*

If the supplier can satisfy the order specified in the RFQ in its entirety, an offer is sent as a response. On the other hand, the supplier may not be able to supply the entire quantity requested in the RFQ by the *DueDate*. In this case the supplier will respond by issuing up to two amended offers, each of which relax one of the two parameters-quantity and 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).
- An ***earliest complete offer*** is generated to reflect the earliest day (if any) that the supplier can deliver the entire quantity requested (due date relaxed).

The agent then has to choose between one of the two offers, not both. In case an agent attempts to order both the partial offer and the earliest complete offer, the supplier will choose the order that arrived first (and ignore the subsequent order). Figure 4 and subsequent description illustrates this. Note, no offer will be made in case a supplier has no components to offer.

All offers made by suppliers are valid for a day and hence require the agent, if interested, to send a confirmation by issuing a purchase order:

Order::=<Order-Id, Offer-Id>

Offer-Id, Order-Id ::= Integer

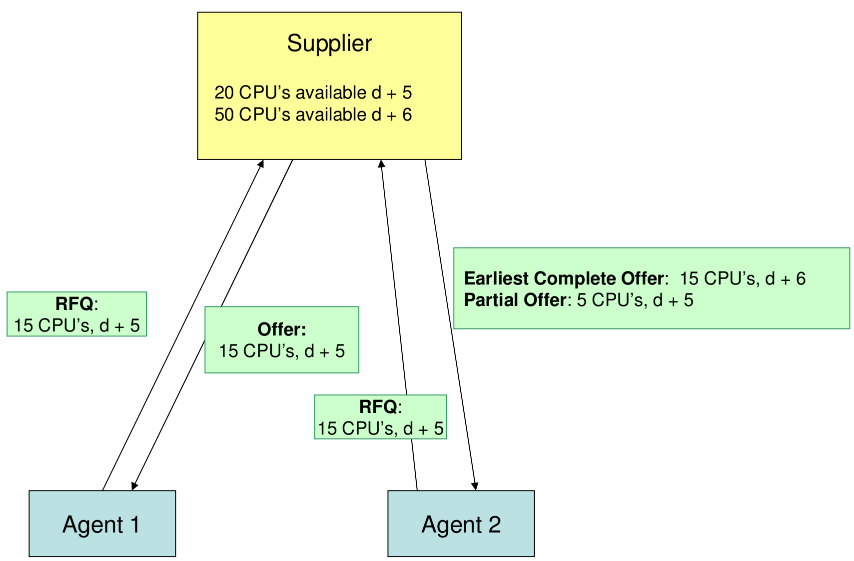

Figure 4: An example of two agents competing for components from one supplier.

Explanation of figure 4: On day d let us assume a supplier forecasts having 20 components available on day d+5 and 50 components available on day d+6. Now, consider the following scenario:

1. Agent 1 sends a RFQ for 15 components to the supplier for delivery on day d+5.

   RFQ <1,c,15,d+5>

2. The supplier, as assumed above, forecasts having 20 of the requested components available and sends back an offer for 15 components to be delivered on day d+5.

   Offer <1,1,15,d+5>

3. Next, the supplier processes an RFQ sent by agent 2 for 15 components also for delivery on day d+5.

   RFQ <10,c,15,d+5>

4. The supplier, having only 5 components available (15 is reserved for agent 1) for delivery on day d+5, and 35 components available for day d+6, sends back:

   a. One *partial offer* for 5 components to be delivered on day d+5.

   Offer<1,10,5,d+5>

   b. One *earliest complete offer* for 15 components for delivery on day d+6.

   Offer <2,10,15,d+6>

5. The agent then may accept one of the two offers, say in this case if the agent believes it can procure the balance 15 components from another supplier, (or for that matter already has an offer from another supplier) then it will choose offer 4a.

5.2 Supplier model

This section explains how the suppliers in the game are modeled. The supplier needs to perform two tasks:

1. Produce and ship components requested by agents.
2. Manage available capacity in order to make offers to agents.

Assumptions

1. Suppliers operate in a make to order basis.
2. If multiple days of production are required to satisfy the order, inventory is carried over. Inventory carrying costs are assumed to be zero.
3. Any excess capacity available on the day of production is used to satisfy future orders in the book. However, the order is only shipped on the due date (or later if not possible to deliver fully that day).
4. A random walk is used to determine the production capacity for each day (constrains the number of units that can be produced each day).
5. Future capacity (forecast) is assumed to be the nominal capacity.
6. If an 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.

5.3 Daily production

Every supplier has a dedicated line with some available capacity, for each component type it supplies. For every component produced by a supplier $C_{nominal}$ denotes the nominal capacity. The nominal capacity is the expected or mean capacity and is used by the supplier for planning purposes. The production capacity C_p on a day, d, is determined by a mean reverting random walk with a lower bound, as follows:

$$C_p(d) = \max\left(0, C_p(d-1) + random(-0.05, 0.05) * C_{nominal} + 0.01 * \left(C_{nominal} - C_p(d-1)\right)\right) \quad (4)$$

Where $C_p(d\text{-}1)$ at the start of the game is $C_{nominal}$. At the start of every TAC day the supplier computes C_p and produces components for as many orders as possible. In the case of 'under capacity' ($C_p < C_{nominal}$), missed orders are scheduled for the next possible day. The delivery is made only when the entire quantity of the order is satisfied. If the delivery can not be made before the game ends, the supplier will deliver as much as it can on the last day.

5.4 Determining available capacity

Suppliers accept orders assuming that they will have $C_{nominal}$ available every day. $C_{free}(d)$ and $C_{ordered}(d)$ denote the free and ordered capacity respectively on any day d. Thus,

$$C_{free}(d) = C_{nominal} - C_{ordered}(d) \tag{5}$$

On any day d the available capacity on some day d+i is given by the sum of all free capacity between the current day and day d+i-1.

$$C_{available}(d, d+i) = \sum_{j=d}^{j=d+i-1} C_{free}(j) \tag{6}$$

An order for delivery on day d+i for n components is accepted if $C_{available}(d+i) \geq n$ or if the components requested are available in inventory and the orders for which the inventory was originally intended still can be delivered on time. In case $C_{available}(d+i) \leq n$, a partial offer promising $C_{available}(d+i)$ is made and an earliest complete offer is generated if there exists some $j>i \wedge j \leq end\text{-}of\text{-}game$ such that

$C_{available}(d+j) \geq (n - C_{available}(d+i))$.

5.5 Supplier pricing

On any day d the offer price of a component for day d+i is given by

$$P(d, d+i) = P_{base}(component)\left(1 - \delta p * \left(\frac{C'_{available}(d, d+i) - qty}{C_{current}(d) * i}\right)\right) \tag{7}$$

where

$P(d+i)$ is the offer price

δp is the price discount factor and has a value 50%

P_{base} is the baseline price of the components (given in figure 3)

$C_{current}$ *(d)* is the suppliers capacity on day d

$C`_{available}$ (d,d+i) is similar to $C_{available}$ given in (6) except C_{free} is calculated using $C_{current}(d,d+i)$ rather than $C_{nominal}$

$C`_{available}\,(d,d+i) = \sum_{j=d}^{j=d+i-1} C'_{free}(j)$ where $c`_{free}(j) = C_{current}(d)-C_{ordered}(d)$

qty is the quantity requested by the order

6 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 1). Agents must bid to satisfy the entire order (both quantity and due date) for the customers to regard the bid.

Customer demand is expressed as requests for quotes (RFQs).

RFQ ::= <RFQ-Id, PC, Quantity, DueDate, Penalty, ReservePrice >

RFQ-Id, PC, Quantity, DueDate, Penalty, ReservePrice ::= Integer

As mentioned earlier customers are classified into three market segments: High range, Mid range, and Low range. For each of these segments, at the start of the day the customer exhibits their demand by issuing RFQs according to the following distribution:

Number of RFQ = poisson (average number of RFQ per day)

The *average number of RFQ per day* issued in each market segment will be varied using a trend that is updated by a random walk given by:

$RFQ_{avg} = min\,(RFQ_{max}, max\,(RFQ_{min}, RFQ_{avg} * trend))$

$$trend = max(T_{min}, min(T_{max}, trend + random(-0.01, 0.01)))$$

The start value of RFQ_{avg} is chosen uniformly in the interval [RFQ_{min}, RFQ_{max}] (see table 1), and the start value of the trend is 1.0. Note, the trend is reset to 1.0 when the random walk hits the minimum or maximum boundaries, i.e. if $RFQ_{avg} = RFQ_{max}$ or RFQ_{min} then trend =1.0 . This is done to reduce the bimodal tendency of the random walk.

For each RFQ, *id* is a unique identifier, *PC* is randomly selected from available types (see Bill of Materials), *Quantity* is chosen uniformly in the interval [Q_{min}, Q_{max}], *DueDate* is current date added to a uniformly chosen number of days in the interval [D_{min}, D_{max}]. Each RFQ also specifies the *MaximumPrice* that a customer is willing to pay. This price termed *ReservePrice,* is randomly chosen in the interval given d in table 1. The customers will not consider any bids with prices greater than the *Reserve Price*. For each customer RFQ*, Penalty* is chosen uniformly in the interval [P_{min}, P_{max}], and is expressed as percentages of the *ReservePrice*. Penalties are charged daily when an agent defaults on a promised delivery date and 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) as they can never be delivered.

Note, agents may ship an order to a customer earlier than the agreed *DueDate* but the customer will pay for it only on the agreed *DueDate*.

The customer only accepts 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 *DueDate* specified in the RFQ,
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. The customer collects all the accepted bids and selects the bid with the lowest price as the winning bid. In case of a tie, the winner is chosen randomly between the tied bids. Note, as mentioned earlier, the winner will be notified at the start of the next day. Also, all agents will be informed about the price interval (min, max) for each type of PC ordered the previous day.

Parameter	**Estimate**
Agent assembly cell capacity	2000 assembly cycles / day
Nominal capacity of all the suppliers assembly lines	500 components / day
Average number of customer RFQs [RFQ_{min}, RFQ_{max}] in the High and Low range	25–100 per day
Average number of customer RFQs [RFQ_{min}, RFQ_{max}] in the Mid range	30-120 per day
RFQ *trend* for customers [$T_{min,}$ T_{max}] (all customer segments)	(0.95 , 1/0.95)
Range of quantities for customer RFQ [Q_{min}, Q_{max}]	1-20
Range of due date for customer RFQ [D_{min}, D_{max}]	Between 3 to 12 days
Range of penalties [$P_{min,}$ P_{max}]	5% to 15 % of the *customer reserve price* per day
Annual Interest rate [I_{min}, I_{max}]	10-20%
Customer Reserve Price	75-125% of base price of the PC components
Storage cost (annual rate) [Sc_{min} , Sc_{max}]	15%-25% of base price of components

Table 1: Parameters used in the TAC SCM game

7 The bank

Agents have an account in the bank and start the game with no money in the account. Money is added to the account when an agent ships a product (to reflect payment by the customer) and is deducted when agents receive supplies from suppliers, or when agents default on deliveries. Agents are allowed to go 'into the red' during the course of the game. A fixed interest rate is chosen uniformly in the range [I_{min} , I_{max}] at the start of game, and is either charged if the balance is in debt or credited if the balance is positive. The interest rate is per year (365 days) and is applied at the end of each game day. All agents have access to bank account information including interest rate.

8 Storage costs on inventory

Inventory (both finished goods and components) will be charged a storage cost *Sc* which is a percentage of the base price of components. The storage cost will be chosen randomly in the range [Sc_{min} , Sc_{max}] (see table 1) 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 held at the end of every day.

9 Periodic reports of market state

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

For each component

- Aggregate quantities produced by all suppliers in the given period
- Aggregate quantities sold by all suppliers in the given period.

Also, a customer demand report is made available to all agents. This report lists the average price and volume sold of each PC type requested during that period. The reports allows an agent to get a handle on both 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 give insight into supply procuring practices of other agents.

10 Format of a game

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.

11 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 day and shown to the game viewers live.

- Bank balance
- Cost of inventory held
- Delivery performance
- Machine utilization

References

1. N. M. Sadeh, R. Arunachalam, J. Eriksson, N. Finne and S. Janson, "TAC'03: A Supply Chain Trading Competition", AI Magazine, Spring 2003.
2. Arunachalam, R., Sadeh, N., Eriksson, J., Finne, N., and Janson, S., "Design of the Supply Chain Trading Competition", IJCAI-03 Workshop on "Trading Agent Design and Analysis", Mexico, August 2003.
3. Arunachalam, R., Sadeh, N., Eriksson, J., Finne, N., and Janson, S., "The TAC Supply Chain Management Game", 2003 CMU-CS-03-184 tech. report.