Modeling Supply Chain Dynamics: A Multiagent Approach*

Jayashankar M. Swaminathan †
Stephen F. Smith ‡ Norman M. Sadeh ‡
Haas School of Business, University of California, Berkeley, CA-94720. †
The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA-15213. ‡

April 1996;
Revised: December 1996; February 1997; April 1997

Abstract

A global economy and increase in customer expectations in terms of cost and services have put a premium on effective supply chain reengineering. It is essential to perform risk benefit analysis of reengineering alternatives before making a final decision. Simulation provides an effective pragmatic approach to detailed analysis and evaluation of supply chain design and management alternatives. However, the utility of this methodology is hampered by the time and effort required to develop models with sufficient fidelity to the actual supply chain of interest. In this paper, we describe a supply-chain modeling framework designed to overcome this difficulty. Using our approach, supply chain models are composed from software components that represent types of supply chain agents (like retailers, manufacturers, transporters), their constituent control elements (like inventory policy), and their interaction protocols (like message types). The underlying library of supply chain modeling components has been derived from analysis of several different supply chains. It provides a reusable base of domain-specific primitives that enables rapid development of customized decision support tools.

Subject Areas: Artificial Intelligence, Decision Support System, Simulation and Supply Chain Management.

1 Introduction

A supply chain can be defined as a network of autonomous or semiautonomous business entities collectively responsible for procurement, manufacturing and distribution activities associated with one or more families of related products. Different entities in a supply chain operate subject to different sets of constraints and objectives. However, these entities are highly interdependent when it comes to improving performance of the supply chain in terms of objectives such as on-time delivery, quality assurance and cost minimization.

*This paper is forthcoming in *Decision Sciences*.

As a result, performance of any entity in a supply chain depends on the performance of others, and their willingness and ability to coordinate activities within the supply chain. A global economy and increase in customer expectations regarding cost and service have influenced manufacturers to strive to improve processes within their supply chains, often referred to as supply chain re-engineering (Swaminathan, 1996). For example, Hewlett Packard's Vancouver division reduced inventory costs by approximately 18 % for HP Deskjet printers through delayed product differentiation (Billington, 1994). Similarly, National Semiconductor has managed to reduce delivery time, increase sales and reduce distribution cost through effective supply chain re-engineering (Henkoff, 1994).

Supply chain re-engineering efforts have potential to impact the performance of supply chains in a big way. Often they are undertaken with only a probabilistic view of the future, and it is essential to perform a detailed risk analysis before adopting a new process. In addition, many times these re-engineering efforts are made under politically and emotionally charged circumstances. As a result, decision support tools that can analyze various alternatives can be very useful in impartially quantifying gains and helping the organization make the right decision (Feigin, An, Connors, and Crawford 1996). In most organizations, reengineering decisions are generally based on either qualitative analysis (such as bench marking) or on customized simulation analysis. This is because complex interactions between different entities and the multi tiered structure of supply chains make it difficult to utilize closed form analytical solutions. Bench marking solutions provide insights into current trends but are not prescriptive. This leaves simulation as the only viable platform for detailed analysis for alternative solutions. However, there are two major problems with building customized simulation models: (1) they take a long time to develop and, (2) they are very specific and have limited reuse. Our aim in this paper is to provide a flexible and re-usable modeling and simulation framework that enables rapid development of customized decision support tools for supply chain management.

It is essential to understand important issues (decision trade-off) and common processes in different types of supply chains to develop a generic, modular and reusable framework. Our framework is based on supply chain studies conducted in the three distinct domains - (1) a vertically integrated supply chain of a global computer manufacturer (Swaminathan, 1994); (2) a Japanese automotive supply chain which is less tightly coupled (Sabel, Kern and Herrigel, 1989); (3) an inter-organizational supply chain in US grocery industry (ECR 1993). These supply chains differ in terms of centers of decision making, heterogeneity in the supply

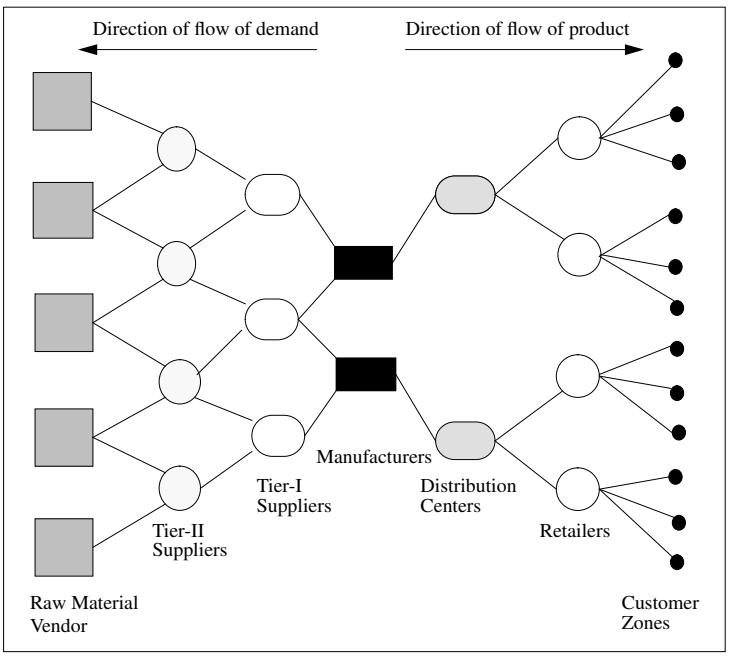

Figure 1: Supply Chain Network

chain and relationship with suppliers. In the supply chain for the computer manufacturer we found that the decision making process was centralized to a great extent, few suppliers were extremely important while others were mainly controlled by the manufacturer and a major part of the supply chain was owned by the manufacturer. In the Japanese automotive supply chain, the manufacturer had a greater control over external suppliers and in some cases partially owned them. However, suppliers made independent decisions many times and, the supply chain involved different companies though all worked according to the guidelines set by the manufacturer. In the grocery supply chain, manufacturers and retailers were equally powerful and sometimes had conflicting interests. The decision making was decentralized, and different organizations (operating under different industrial environments) were part of the same supply chain.

Despite these differences, we found that there are a number of processes which are common to these supply chains. We have identified these processes and have developed a library of software components for modeling them. The library consists of two main categories- *structural* elements and *control* elements. *Structural* elements (like retailer,

distribution center, manufacturer, supplier and transportation vehicles) are used to model production and transportation of products. *Control* elements are used to specify various control policies (related to information, demand, supply and material flow) that govern product flow within the supply chain. Given this base of primitives, an executable simulation model of a given supply chain model is constructed by instantiating and relating appropriate structural and control elements. Our framework allows development of models to address issues related to *configuration*, *coordination* and *contracts*. *Configuration* deals with issues related to the network structure of a supply chain, based on factors such as leadtime, transportation cost and currency fluctuations; *Coordination* deals with routine activities in a supply chain such as materials flow, distribution, inventory control and information exchange; *Contracts* control material flow over a longer horizon based on factors such as supplier reliability, number of suppliers, quantity discounts, demand forecast mechanisms and flexibility to change commitments.

Multi-agent computational environments are suitable for studying classes of coordination issues involving multiple autonomous or semi-autonomous optimizing agents where knowledge is distributed and agents communicate through messages (Bond and Gasser, 1988). Since supply chain management is fundamentally concerned with coherence among multiple decision makers, a multi-agent modeling framework based on explicit communication between constituent agents (such as manufacturers, suppliers, distributors) is a natural choice. We model *structural* elements as heterogeneous agents which utilize *control* elements in order to communicate and control flow of products within the supply chain. Our approach emphasizes models that capture the locality that typically exists with respect to the purview, operating constraints and objectives of individual supply chain entities and, thus promotes simultaneous analysis of supply chain performance from a variety of organizational perspectives. The modular architecture of our framework enables one to develop executable models for different situations with limited additional effort.

A typical supply chain faces uncertainty in terms of supply, demand and process. Our framework reduces the effort involved in modeling various alternatives and measuring their performance through simulation under different assumptions about uncertainties. This eases the ability of decision makers to quantitatively assess the risk and benefits associated with various supply chain re-engineering alternatives. In this paper, we describe our framework in its current state and provide examples to demonstrate how issues relevant to supply chain management can be analyzed using it. A software application using some of the concepts

from this framework has been developed at IBM.

The rest of this paper is organized as follows. Section 2 reviews existing research and approaches. In section 3, we describe our multi-agent framework in greater detail. In section 4, we identify the key elements required to model supply chain dynamics. In section 5, we present a cross docking prototype from the grocery chain industry. In section 6, we describe a full scale application developed for IBM asset managers and, we conclude in section 7.

2 Literature Overview

Bench marking efforts aimed at identifying new trends and philosophies in supply chain management based on comparative analysis of current practice in different countries and different sectors of industry include those reported in Hall(1983), Helper(1991), and, Lyons, Krachenberg and Henke(1990). Lee and Billington (1992) provide an insightful survey of common pitfalls in current supply chain management practices. Some studies indicate that buyer-supplier relationships are becoming more dependent on factors like quality, delivery performance, flexibility in contract and commitment to work together, as opposed to traditional relationships based on cost (Helper, 1991). Electronic Data Interchange (EDI) and Distributed Databases have been identified as important technological advancements that may benefit supply chain performance in a significant manner (Srinivasan, Kekre and Mukhopadhyay 1994). While providing general guidelines and identifying elements of best practice, the bench marking approach has been of limited help to managers who are looking for specific quantitative solutions to every day problems.

On the analytical front, research on multi echelon inventory problems has a long history (Clark, 1962; Clark and Scarf, 1958; Svoronos and Zipkin, 1991). A multi echelon system is one in which there are multiple tiers in the supply chain. This line of work typically assumes centralized control of the supply network, thus overlooking the possibility of decentralized decision making. More recent supply chain models in this area also include Cohen and Lee(1988), Cohen and Moon(1990), and, Newhart, Scott and Vasco(1993) where deterministic scenarios are considered and a global optimization problem is formulated using mixed integer programs. Lee and Billington (1993), and, Pyke and Cohen (1993, 1994) consider stochastic environments and provide approximations to optimal inventory levels, reorder intervals and service levels. Arntzen *et al.* (1995) develop an elaborate model for global supply chain management for Digital Equipment Corporation. The above work has

contributed in a significant manner to managerial decision making. However, these models are limited in handling issues related to dynamics of supply chains and focus exclusively on global performance measures.

The use of simulation as a vehicle for understanding issues of organizational decision-making has gained considerable attention and momentum in recent years (Feigin *et al.* 1996; Kumar, Ow and Prietula, 1993; Malone, 1987). Towill, Naim and Wikner (1992) use simulation techniques to evaluate effects of various supply chain strategies on demand amplification. Tzafestas and Kapsiotis (1994) utilize a combined analytical/simulation model to analyze supply chains. Swaminathan, Sadeh and Smith (1995) utilize simulation to study the effect of sharing supplier available-to-promise information. Given the utility of this approach, there is a need for tools that can facilitate rapid development of simulation models. Since simulation models in general have limited reuse, the above tools should provide an environment where re-usable software components are essentially combined to construct simulation models for different problems. Simulation software is more prevalent in the area of business process re-engineering in a broader sense. Swain(1995) provides an extensive survey of commercial simulation software packages available for process analysis. Among them software packages like Extend+BPR, Ithink, SIMPROCESS-III and Work-Flow Analyser allow modeling and analysis of business processes. Currently there is no commercial simulation software that provides domain specific primitives for modeling and analyzing supply chain coordination problems. In addition, most of the above software systems are built around simple control mechanisms for processing events such as first in first out (FIFO) queues. However, supply chain interactions typically involve more sophisticated control mechanisms. For example, when an important order comes in, it may have to be processed first, ahead of other orders. Also processing of an item may involve more than just waiting at the service center for some time. For example, when an order is processed, components may have to be assembled and that could in turn trigger some events based on their inventory position. Decision rules may have to be used at various points when events are processed. In order to model problems related to supply chain management or for that matter any particular domain one requires specialized primitives. Our aim in this paper is to provide a modular and re-usable framework with primitives that allow development of realistic supply chain models.

3 Multi-Agent Framework

The approach in this work has been to utilize a multi-agent paradigm for modeling and analysis of supply chains. Multi-agent computational environments are suitable for studying a broad class of coordination issues involving multiple autonomous or semi-autonomous problem solving agents (Bond and Gasser, 1988). Knowledge-based multi-agent systems have been found useful in many applications related to manufacturing including scheduling, vehicle routing and enterprise modeling (Kwok and Norrie, 1993; Pan, Tanenbaum and Glicksman, 1989; Roboam, Sycara and Fox, 1991; Sadeh, 1994; Smith, 1989). In this work we have extended the use of multi-agent paradigms to the domain of supply chain management. We identify different agents in the supply chain and provide each agent an ability to utilize a subset of control elements. The control elements help in decision making at the agent by utilizing various policies (derived from analytical models such as inventory policies, just-in-time release, routing algorithms) for demand, supply, information and materials control within the supply chain. Our analysis is based on discrete event simulation of the various alternatives and control policies. Combination of analytical and simulation models makes our framework attractive to study both the static and dynamic aspects of problems.

We have defined a generic agent which is then specialized to perform different activities within a supply chain. For example, a manufacturing agent is different from a distribution agent or a transportation agent. Specialized agents correspond to *structural* elements identified in the supply chain library that are involved with production and transportation of products within the supply chain. Different agents in our framework communicate with each other through messages. Incoming messages are selected by each agent based on an event selection mechanism such as first come first served (FCFS). Each message type has a message handler or a script that determines how the message will be processed. The message handler is parametrized by the control policies that are used by the agent. For example, the message handler corresponding to a *request for goods* message performs the following actions.

1. Check if the product is available in stock. If that is the case then the demand is satisfied and inventory on-hand is updated else the demand is backlogged and the status of backlogged demand is updated.

2. The inventory control policy (say a base stock policy) is invoked.

3. The inventory control policy generates a *request for goods* message for the supplier of the product based on inventory on-hand and backlogged demand. It may utilize supplier capacity information based on agreements for information sharing with the supplier.

4. If outgoing messages are generated they are queued up in the global message queue with a time stamp for activation.

Since our framework is based on a discrete event simulator, agents are activated based on the time of activation of incoming messages. There is a global list of incoming messages for all agents, sorted in terms of time of activation, and the agent which has the earliest message is processed next. The simulation clock is advanced to the activation time. Agents that did not process a message at a given time instant retain their state and knowledge about other agents in the next time instant. Simulation continues for the total simulation time specified by the user at the beginning of the simulation.

In the next subsection we introduce the generic agent architecture. Subsequently, we define various messages in our framework.

3.1 Agents

Agent descriptions provide an ability to specify both static and dynamic characteristics of various supply chain entities. Each agent is specialized according to its intended role in the supply chain (for example manufacturer agents, transportation agents, supplier agents, distribution center agents, retailer agents, end-customer agents). An agent is defined by the following set of characteristics at a given time instant.

- S_i - Set of attributes that characterize its (simulated) state at a given instant of time. State attributes include base information about an agent's processing state (for example, current product inventories, different costs associated with production, financial position). Associated with each aspect of local state are methods for accessing and (in the case of dynamic parameters) updating current values. Dynamic parameters change over time either as the result of internally triggered events (for example, when material gets transferred from work-in-process inventory into finished-goods inventory) or as a result of interactions with other agents (for example, receipt of an order from a customer, shipment of an order to a customer, payment for an order delivered to a customer).

- D_i - Knowledge at agent i about other agents. Since each agent is locally defined, it will typically have only an incomplete view of the state and actions of other agents. This includes information about the past performance of the different agents. These values may also be updated dynamically during simulation. For example, when it is known that a reliable supplier defaults often in terms of due date then that agent's reliability factor is updated accordingly.

- IC_i - Set of interaction constraints that define the agent's relationship with other agents in the supply chain. Each agent description designates the set of agents with which it can interact, and for each, indicates (1) its relationship to this agent (customer, supplier), (2) the nature of agreement that governs the interaction (production guarantees, agreement length) and inter-agent information access rights (which aspects of that agent's local state are accessible for consultation during local decision-making). All the information about other agents that is available without message transfers is controlled by the real-time information control policy (described in section 4.2.5).

- Q_i - Priorities of agent i. These help in sequencing incoming messages for processing.

- PM_i - Vector of performance measures of agent i.

- I_i - Set of incoming messages at agent i.

- O_i - Set of outgoing messages at agent i.

- c_i - Incoming message that is chosen for processing by agent i.

- ϕ_i- Set of control elements available at agent i. A control element is invoked when there is a decision to be made while processing a message. For example, in order to determine the next destination on a transportation vehicle a route control element will be invoked.

- $M_i(c_i)$ - This defines the message processing semantics for message type c_i at agent i. Message handling routines may use one or more control elements which processing a message. For example, when a *request for goods* message is processed it invokes a inventory control policy. In some cases, more than one control element may be used. For example, an information control element may be invoked to obtain capacity information from the supplier agent before invoking the inventory control policy.

- $P(D_i, S_i, I_i, Q_i)$ - A selector function that chooses and sequences a set of incoming messages based on domain knowledge, current state and the priorities of agent i. For example, when a manufacturer has orders from two customer agents then this function would determine the sequencing rule based on the priority given to each customer agent. Sequencing becomes important when the manufacturer does not have enough inventory to satisfy all the orders.

The sequence of events that occur at each agent that processes incoming messages is as follows (refer Figure 2). We will explain the processing of a message taking the example of a retailer agent. Each type of agent is defined with respect to a specific set of goals which determine commitments and control elements that it uses while interacting with other agents. For example, the goal of the retailer is to reduce the turn-around time that the customer experiences while keeping the inventory costs under control. Commitments of a retailer agent might include service constraints such as 98% of orders fulfilled within a day for top priority customers. In order to fulfill such commitments, a retailer agent may utilize advanced inventory control policies and real-time information sharing with manufacturers. Performance measures of the agent as well as the above commitments influence priorities Q_i of the agent. These priorities determine the sequence in which incoming messages I_i are processed. For example, the retailer agent may prioritize customers according to an A-B-C classification thereby, sequencing the "A" customer order before others when there is more than one outstanding order. The first message in the sequence c_i is analyzed for the type. It could be a material, information or financial message (as described in section 3.2). Each message type has a message handler $M_i(c_i)$ that specifies a sequence of operations to be performed and may involve usage of one or more control policies as explained in the example at the beginning of this section.

The message handling routines for the same message type may be different in different agents. For example, when a *goods delivered* message is encountered in a standard distribution center, materials are stored in a storage location, whereas in a cross dock, materials are sorted by destination and outgoing truck loads are updated. If any of the outgoing vehicles are completely loaded then, an appropriate *goods delivered* is posted for the receiving agent. Control elements are triggered by the message handler $M_i(c_i)$ at relevant decision points (for example, a reordering decision or a routing decision). Message handling routines may also update the internal state and the domain knowledge, and, generate one or more

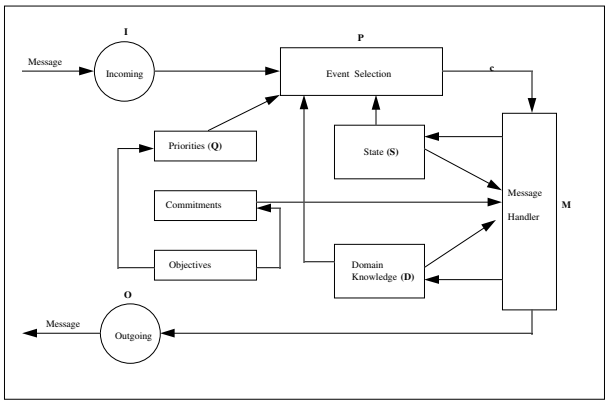

Figure 2: Agent Architecture

outgoing messages. For example, when goods are received at the retailer agent, inventory of corresponding items are updated and a payment message may be posted for the agent supplying the materials. If the materials came in later than promised then the reliability of the agent supplying the material is updated in the domain knowledge. Once the message processing operations have been completed, local performance measures of the agent PM_i and the global performance measures are updated. For example, when goods are received at the retailer agent, the inventory levels will be recorded so that average inventory holding costs can be determined at the end of the simulation. Outgoing messages have the address of the destination as well as the time that they will be activated at that agent (which maybe different from the current time due to delays). This process continues at an agent till there is no active incoming message at the given instant of time.

3.2 Interaction Protocols

A basic set of *message classes* define the types of interactions that can take place within the network. All message classes share specific common attributes, including the (simulated) time at which they are posted, the time they get activated, the posting agent and the recipient agent. Associated with each message class, are message handlers that are parametrized by the control policy used by the agent and, in essence, define message processing semantics.

As indicated earlier, this may depend on the type of agent where the message is processed.

We recognize three broad categories of message classes, each associated with the simulation of a specific type of flow through the supply chain:

- *Material flows*: Messages in this category relate to delivery of goods by one agent to another. The processing semantics associated with material delivery messages minimally dictate adjustment to inventories of the posting and recipient agents by the quantity specified in the message. However, it can also trigger messages relevant to other supply chain flows (cash transactions) as well as local processing activities (determination of whether all the components required to initiate the assembly of a product are now available). Material delivery messages can be either sent directly by a supplier agent to a consumer agent (in cases where simulation of transportation delays and costs are not relevant) or may involve an intermediate transportation agent.

- *Information flows:* This category of messages model exchange of information between supply chain agents. It includes *request for goods* messages (flow of demand), *capacity information* (communication of expected available capacity), *demand-forecast information* (communication of demand forecasts) and *supply-related information* (expected delivery dates). Other messages that fall in this category include *order cancellation* messages and *order modification* messages (modified quantity or due date).

- *Cash flows:* The final category of message classes concern the movement of capital through the supply chain. This category includes a *payment* message sent by customer agents to their supplier upon delivery of goods.

3.3 Performance Measures

One of the objectives of developing an integrated framework is to provide an ability to simultaneously observe global and local performance of the supply chain. Empirical studies have shown that sometimes taking a global perspective may be harmful to some of the entities in the supply chain (Cash and Konsynski, 1985; Swaminathan, Sadeh and Smith 1995). In our framework we separate local performance (PM_i) from the global performance measures (GPM). A global performance measure may be an appropriate yardstick for an intra organizational supply chain (most of the entities belong to the same organization)

however, local performance becomes an important measure for inter-organizational supply chains.

Supply chain performance measures can be classified into two broad categories. *Qualitative* performance measures such as customer satisfaction, integration of information and material flow and, effective risk management. *Quantitative* performance measures relate to cost minimization, profit maximization, fill-rate maximization, customer response time minimization, supplier reliability and lead time minimization. In our framework, we consider only quantitative performance measures. We provide the capability for analysts to monitor appropriate performance measures (either local or global or both) depending on the situation. It should be noted that there is a very strong link between goals of the agent in terms of the performance measures PM_i and priorities Q_i of an agent. These priorities determine the sequence in which incoming messages are selected and in some sense drive the simulation.

Our framework is based on simulation and the performance is dependent on the starting condition and the length of simulation. Repeated simulations under different starting conditions should be performed in order to obtain robust output. Many times supply chain decisions are made under uncertainty about future and our framework provides the ability to model supply, demand and process uncertainty within the supply chain and perform a detailed risk analysis. Some of the *configuration* related issues involve analysis of long term decisions and potential risks associated with them. Our framework can be utilized while making those decisions by developing different simulation models for alternative configurations and evaluating them while using the same set of input parameters. Comparison of the performance of alternative configurations provides the manager with information about the expected benefit from each alternative. The manager would choose one among the various alternatives based on their estimated cost, their measured performances and other managerial criteria that could not be modeled in the simulation. In addition to providing all the advantages of utilizing simulation, our framework enables the user to model a broader set of supply chain issues under a reduced development time which is particularly useful while performing risk analysis prior to supply chain re-engineering.

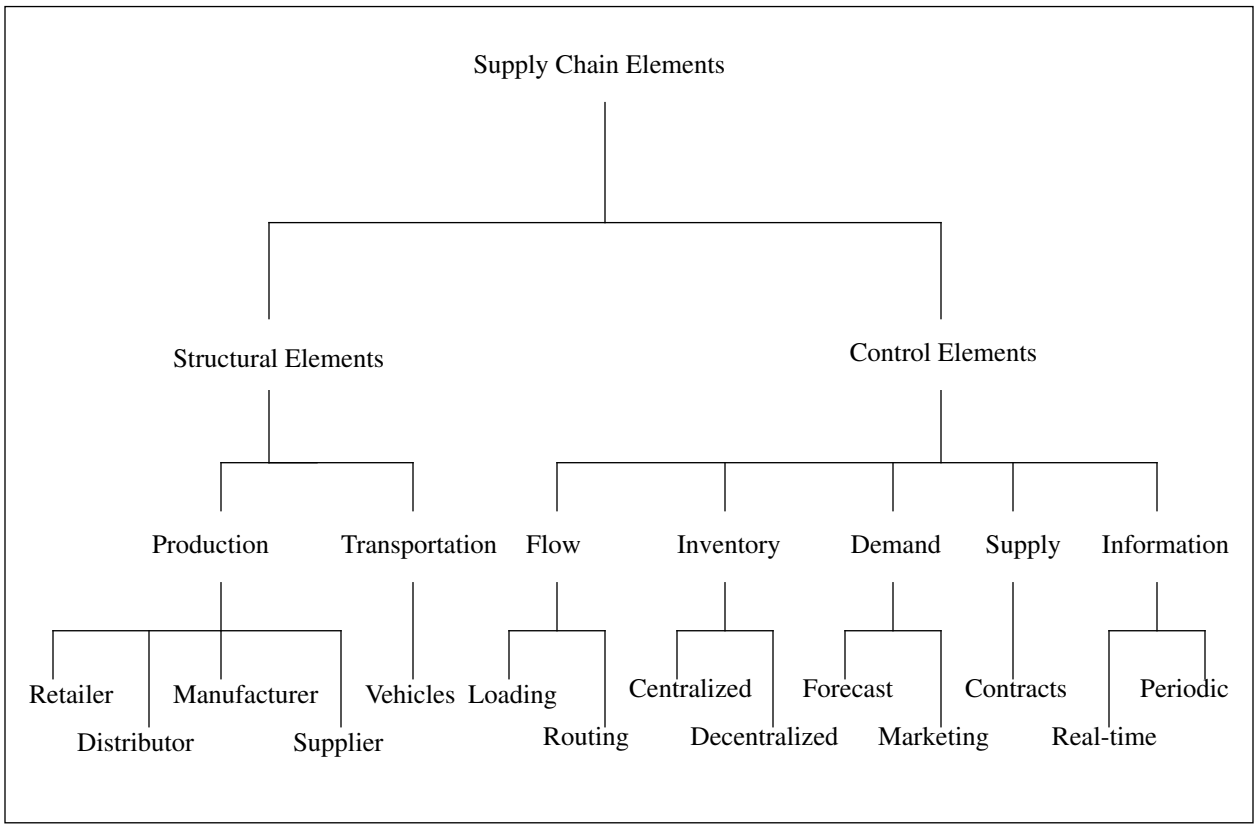

Figure 3: Structure of Supply Chain Library

4 Supply Chain Library

Supply chain dynamics can become complicated to model due to presence of heterogeneous entities, multiple performance measures and complex interaction effects. The variety of supply chains poses a limitation on reusability of processes across them. For example, a supply chain could be highly centralized and have most of the entities belonging to the same organization (like IBM integrated supply chain) or could be highly decentralized with all the entities being separate organizations (like a grocery supply chain). As a result, it is a difficult task to develop a set of generic processes that capture the dynamics of supply chains across a wide spectrum. In this section, we present a classification of library of software components which enables modeling and analysis of a large variety of problems though it is not exhaustive by any means.

We classify different elements in the supply chain library into two broad categories-

Structural elements and Control elements (refer Figure 3). Structural elements (modeled as agents) are involved in actual production and transportation of products and Control elements help in coordinating the flow of products in an efficient manner with the use of messages. Structural elements correspond to agents and control elements correspond to the control policies in our framework. Structural elements are further classified into two basic sets of elements namely, Production and Transportation elements. Control elements are classified into Inventory Control, Demand Control, Supply Control, Flow Control and Information Control elements.

4.1 Structural Elements

As indicated earlier, structural elements are involved in production and transportation of products. Strategic placement of these elements constitutes major issues relating to supply chain configuration. In the following subsections we briefly describe each of the structural elements.

4.1.1 Production Agents

Production agents use inventory control elements for managing their inventory, contracts with downstream entities for supply control, flow control elements for loading and unloading products, forecast elements for propagating demand forecasts to the downstream entity and may use information control elements with other entities in the supply chain.

- Retailer: A retailer is where customers buy products. The main focus here is on reducing the cycle time for the delivery of a customer order and minimizing stock-outs. The above goals define the objectives and priorities of this agent which are used while sequencing incoming messages. When a customer order for a product is received, it is determined which product is being ordered. The product is packed and shipped to the customer if it is available as finished good inventory, or else the order is added to a queue (for the particular product) according to its priority (if the priority of all the orders are same then it is FIFO (first-in-first-out)). When the product is delivered from the distribution center or from the manufacturing (it is possible that some products may come from the manufacturing plant while others could come from the distribution center) plant, the order is removed from the queue and product is packed and shipped to the customer. Many times, orders may be placed for multiple

products in which case the processing becomes more complicated. Marketing elements (described in section 4.2.2) are used for controlling demand generated by customers.

- Distribution Center: A distribution center is involved in receiving products from the manufacturing plant and either storing them or sending them right away (cross-dock) to the retailer. The main focus here is to reduce the inventory carried and maximize throughput. In a standard distribution center products come in from the manufacturing or supplier plants. They are unloaded and stored in the storage area. When orders come from the retailer, relevant products are removed from the storage area (if the buffer has them or they wait till the products arrive into the buffer) and are sent to the appropriate loading dock where they are loaded and sent to the destination. As opposed to a standard distribution center, in a cross-dock there is no inventory storage. Products are unloaded from one transportation vehicle and are directly loaded onto outgoing vehicles to different retailers.

- Manufacturing Plant: A manufacturing plant is an agent where components are assembled and a product is manufactured. In general, orders come from the distribution center but they could also come from the retailer (when there is a cross-dock or the supply chain does not have a distribution center). The main foci here are on optimal procurement of components (particularly common components) and on efficient management of inventory and manufacturing process. Each product has an associated bill of materials (BOM). Manufacturing can be based on either a "Pull" or "Push" mechanism. In a Pull system, product is made only when an order is received for it; in a Push system, products are built based on demand forecast.

- External Suppliers: An external supplier agent models external suppliers. These suppliers could be a manufacturing plant or assembly plant or could have their own supply chain for production. However, we model all these situations through a single agent because the parent organization has no direct control on their internal operations. Supplier agents supply parts to the manufacturing plant. They focus on low turn-around time and inventory. Their operation is characterized by the supplier contracts which determines the leadtime, flexibility arrangements, cost-sharing and information-sharing with customers.

4.1.2 Transportation Agent

- Transportation Vehicles: Transportation vehicles move product from one production agent to another. Each vehicle has associated characteristics in terms of capacity and relative speed. Vehicles use flow control elements in order to load and unload the products as well as to determine the route. The route taken by the vehicle depends on the state of the vehicle (which contains information on destination of products that have been loaded). Using distance to the next destination from the current destination, the time needed to reach the next destination is obtained. At that time, products (destined for that production agent) are unloaded and other products may get loaded.

4.2 Control Elements

Control elements facilitate production and transportation of products within the supply chain. Choice of appropriate control elements is the objective of problems related to supply chain contracts and supply chain coordination. Here is a brief description of control elements currently defined.

4.2.1 Inventory Control

Inventory control elements are an integral part of any supply chain. They control flow of materials within the supply chain. They are mainly of two types - Centralized and Decentralized control.

- Centralized Control: These elements control the inventory at a particular production element while taking into account the inventory levels in the supply chain as a whole. A typical example is inventory control based on *echelon* inventory. According to this policy, inventory control is applied while considering the total inventory upstream, also called *echelon* inventory. Thus, the order-upto levels are set according to *echelon* inventory levels. An important requirement for implementing a centralized inventory policy is the ability to access information on inventory levels at other entities in the supply chain.

- Decentralized Control: These elements control inventory at a particular production element by considering inventory levels at that entity in the supply chain. Typical examples of these kinds of policies are- order-upto or base stock policy, MRP based

ordering (with no information about inventory status at other agents) and (Q,R) or (s,S) policy. These policies are also used in centralized control though inventory levels in those cases are calculated based on *echelon* stock. In a base-stock policy, orders are placed as soon as the inventory level reaches below the base-stock level in order to bring it back to that level. In MRP based ordering, the requirements are based on the MRP explosion (considering the forecasts as exact) and in (s,S) [(Q,R)] policy, ordering is done when the inventory levels goes below s [is equal to R] and orders are placed so that inventory is brought upto S [Q+R].

4.2.2 Demand Control

The demand process within a supply chain is sustained through actual and forecasts (these are modeled as messages in our framework). Orders contain information on - types of products which are being ordered, the number of products that are required, the destination where the product has to be shipped, and the due date of the order. Two important demand control elements are:

- Marketing Element: One of the important aspects of product management is how well the product is marketed to consumers. There are numerous ways to increase demand for a particular product. These include advertisements, discounts, coupons and seasonal sales. The marketing element provides a mechanism that can trigger additional demand for products. Increase in demand could be seasonal, random or permanent. This element allows us to capture marketing strategies that might be used in the supply chain. We restrict the usage of these elements only at the retailers because these elements can have a direct impact on demand experienced by the supply chain (in some sense we capture the effect on end-consumers only). Demand can be influenced by other agents as well without utilizing these elements (like a supplier agent providing bulk rates to increase the purchases made by the manufacturer).

- Forecast Element: Forecast elements determine how forecasts are generated within the supply chain and how they evolve over time. In a "Push" system, forecast evolution plays a very important role because manufacturing decisions are based on demand forecasts. Greater forecast inaccuracy leads to greater mismatch between products demanded and products produced, and as a result leads to higher inventory costs. In a "Pull" system, products are built-to-order still forecast accuracy plays an important

role in materials procurement and capacity planning.

4.2.3 Supply Control

Supply Control elements dictate terms and condition for delivery of the material once orders have been placed. Contractual agreements are the only form of supply control element that we have identified. Contracts contain information on the price of the material, length of the contract, volume to be purchased over the contract period, penalty for defaulting, leadtime to get the product once the final order has been placed, the amount of flexibility that the buyer has in terms of updating demand forecasts over time (often referred to as *flexibility* offered by the supplier) and types of information control that could be used. Supply contracts may differ in characteristics and rigidity depending on whether supplier of the product belongs to the same organization or not. Transfer pricing mechanisms are employed while dealing with internal suppliers (this could be thought of as a form of centralized supply control).

4.2.4 Flow Control

Flow control elements coordinate flow of products between production and transportation elements. Two types of flow control elements are:

- Loading Element: Loading Elements control the manner in which the transportation elements are loaded and unloaded. This control is different based on the type of the production element where products are loaded or unloaded. For example, loading and unloading operations require different specifications depending on whether the production element is a standard distribution center or a cross-dock. This control element is located in the corresponding production element.

- Routing Element: Routing elements control the sequence in which products are delivered by the transportation element. The route taken by the transportation vehicle depends to a great extent on the destination of products that it is carrying. So, the routing is dynamic in that sense. The route can be decided in a centralized or a decentralized manner depending on how much information is available about destination of other transportation elements.

4.2.5 Information Control

Information control elements are essential for coordination within the supply chain. Two types of information flow are:

- Directly Accessible: Directly accessible information transfer refers to the instantaneous propagation of information. For example this could be information on inventory levels, capacity allocations, machine breakdowns etc. at other production elements or the routes to be taken by other transportation elements.

- Periodic: Periodic information updates may be sent by different production and transportation elements to indicate changes in business strategy, price increase, introduction of new services or features in the products, introduction of new production elements etc. Periodic information is sent to all the entities in the supply chain in the form of messages, as opposed to real-time information, which is explicitly agreed upon in the supply control element.

The above defined set of elements along with the Customer agent that generates demand for the system constitute our framework.

5 A Cross-Docking Prototype

In this section, we provide a detailed example to illustrate how a model is developed utilizing the primitives in our framework. We describe a model from the grocery chain industry which was developed to understand tradeoffs associated with operating a distribution center as a cross dock. One of the major concerns in the grocery chain industry is to try to reduce inventory within the supply chain. A cross-docking center differs from a standard distribution center in that inventory is never stored there. Inventory comes on one truck and leaves on another based on its destination. A cross-docking center only helps in sorting and shipping inventory to the correct destination. As a result, in a cross-docking environment, it may take more time to replenish orders at the retailer because inventory is not stored at the distribution center. A cross-docking environment is also information intensive because all that information is used in effectively sorting and shipping products. The question of interest here is to understand the tradeoff between inventory and service in the alternative arrangements and additional information requirements for a cross dock. Since a grocery chain typically consists of different organizations, it is all the more important to understand

the effect of any change in the supply chain on the different entities. As a result, tracking individual performance is as important as tracking the global performance measure. We track the inventory and as well as customer service measure locally as well as globally. We first develop a simple model and illustrate how it fits in our framework. Subsequently, we compare the development process of this model using our framework with the development process using a standard simulation language.

5.1 Model Building Process

We consider a supply chain with three retailer agents, one distribution agent, three manufacturing agents and one customer agent. Each of the three manufacturing agents produce one unique product. The state of these agents is defined by finished goods inventory and outstanding orders. The customer agent generates demand for the three retailer agents for a mix of these three products. The state of the customer agent consists of only orders that have not been delivered as yet. Each retailer agent stocks inventory of all three products and operates under an inventory control policy such as base-stock for each product. The state of the retailer agents is determined by the inventories associated with each of the three products and the outstanding orders from the customer. We assume that these products can be made by the manufacturers without purchase of any components and as a result, the supply chain ends there. In a model with a standard distribution center, orders (messages) from retailer agents would be stored at this intermediate location whereas in a cross-docking environment we assume that the orders go directly to the manufacturer. We also assume that products are transferred in truck loads and the release policy at the manufacturing agents is a batch policy. The state of the standard distribution center is characterized by similar attributes as a retailer agent. However, a cross dock is characterized by inventory of incoming and outgoing products. We have neglected transportation issues related to coordination of trucks by assuming that trucks are available in plenty. A more detailed model could be developed using transportation agents.

The interaction constraints at each agent are limited to specifying the buyer-supplier relationships. We restrict our attention to only inventory control policies. The customer agent generates product demands based on the demand control policy employed which basically determines the type of demand (periodic or continuous) as well the nature (deterministic or stochastic). The *request for goods* message generated has the address of the retailer as well as the due date by which it is required. Incoming messages from the retailers have

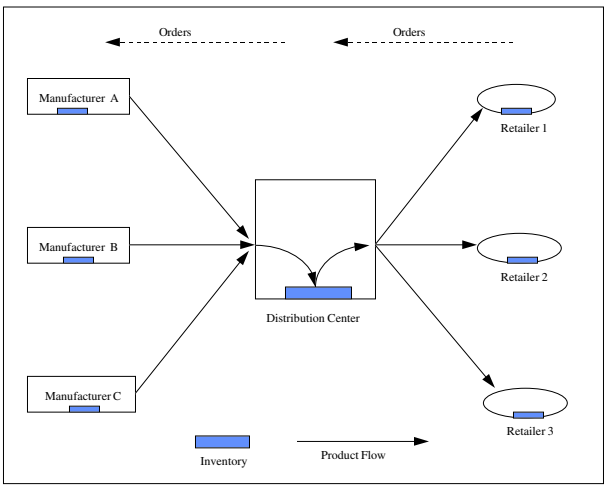

Figure 4: Standard Distribution Center

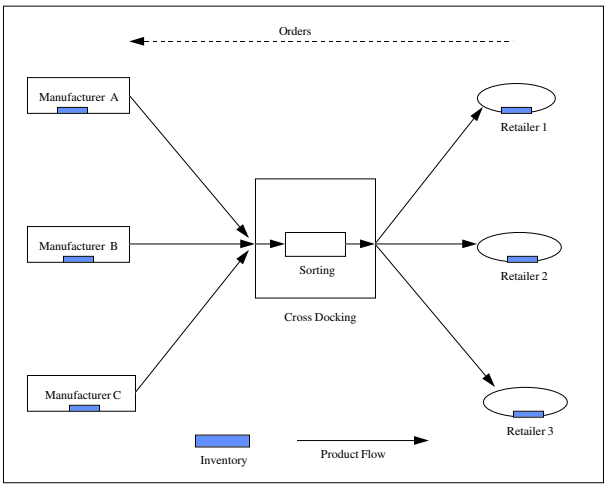

Figure 5: Cross Docking

the due date as well as the current time. Statistics are maintained on the late orders as a performance measure. Each retailer agent processes an incoming message based on its type. It is either a *request for goods* or *goods delivered*. If it is *goods delivered*, then the inventory level is adjusted accordingly, outstanding orders are taken care of and messages are sent to the customer agent. If it is a *request for goods*, then the inventory position is checked. If inventory is available the order is fulfilled and future orders placed based on the inventory control policy. If inventory is not available then the order is made outstanding or lost based on whether demand is backlogged or not. Inventory position is tracked at each instant of time at each agent and is maintained as a performance measure. The standard distributor agent stores inventory and replenishes them from manufacturers. The prime difference being that products are shipped to retailers in truck loads. So, a number of *goods delivered* messages are collected together before being sent to the retailer. In a cross-docking mode, no inventory is stored. Each of the manufacturing agents maintain finished goods inventory and produce in batches. Shipments to distributor agent are made in truck loads. With the above simple model it is possible to analyze some of the trade-off in the alternative arrangements for distribution. Moreover, the results bring out benefits for different entities in alternative arrangements and provide a basis for negotiating cost and benefit sharing in the supply chain. Other variations of this supply chain can be easily analyzed as well. Suppose, if we wanted to analyze the effect of changing the inventory control policy at the retailer agent, we just need to specify a different control policy (from the set of existing control policies in our supply chain library) at the agent and simulate again. Similarly, we can study the effect of introducing one more retailer or one more manufacturer by introducing an agent of that type, defining its relationship to other agents and simulating the new supply chain.

5.2 Comparison with Conventional Approaches

In principle, one could implement our cross-docking model in any conventional simulation language (e.g., GPSS, SIMAN). However, the model building task would be quite different. The primitives provided by conventional simulation languages are much lower-level (like queues) and they are typically defined as extensions to standard procedural programming language constructs. Hence, development of a supply chain model becomes a conventional programming task, and the model just described would require considerable programming expertise and effort. With our approach, in constrast, the view is that models are developed

without resorting to significant programming effort, through use (and re-use) of higher-level modeling primitives which encapsulate important components (or building blocks) of supply chain models. Our vision is that simulation models are configured (not programmed) by selecting, instantiating, and composing sets of components to form an executable simulation model, without the need for extensive programming expertise. Thus, our framework could be utilized directly by supply chain managers who are faced with specific configuration, contracting or coordination issues. Such models, once built, are not that different from any simulation model, and all the benefits of customized simulation models are retained.

To illustrate the above point, consider the 3 retailer, 3 manufacturer model just discussed. Within our framework, the model is obtained by (1) creating instances of "structural" primitives (like manufacturer, retailer etc.), (2) connecting agents to one another thereby defining their relationships and flow of products and information, (3) associating appropriate inventory control policies and coordination protocols at different agents, and (4) setting the demand characteristics and time for simulation. On the other hand, consider development of just a model of a manufacturer with the desired inventory control policy within a conventional simulation language. To start with the develop would need to define incoming and outgoing queues for orders, buffers for storing finished goods and raw material inventory, and delay processes for modeling the manufacturing process. Using the above data structures a software module is created that can replicate the production process. In addition, another module needs to be written for inventory control. The above software modules along with other modules (such as processing cash flows, updating information about buffers and queues, updating information received from other entities) would need to be integrated to form a manufacturer which in turn is integrated with a discrete event simulation engine.

On a relative scale the time taken to instantiate a manufacturer with our framework would in the order of minutes whereas an experienced programmer could develop the module in a couple of days utilizing standard programming tools. In addition, if one wanted to develop a complex supply chain model, the user could develop that in an hour or so using the framework, whereas, developing that model from simulation primitives could take a few months. Since most of the elements in the framework are software objects developed in a simulation language the amount of lines of code as well as speed of execution of simulation remains almost the same. In some cases, one could develop models using simulation primitives which are marginally more efficient in terms of speed and software size. However,

the main advantage of utilizing our framework is that the development time is drastically reduced and the programming effort is minimized. The software behind the library elements of the framework is designed for reuse in the development of new models.

6 A Full Scale Application

Our framework was mainly motivated to address problems faced by managers in charge of supply chain re-engineering efforts in large organizations. As indicated earlier, most of the re-engineering efforts are undertaken with only a probabilistic estimate of the future. As a result, risk and benefits associated various alternatives need to be evaluated before an alternative is chosen for implementation. IBM researchers have developed detailed simulation models that have provided management with many insights and enabled the supply chain re-engineering efforts [Feigin *et. al.* (1996)]. Such simulation systems take a long time to develop, prototype and implement (typically range from 12 to 20 man months). In addition, it was often difficult to utilize the same system for similar re-engineering efforts within the organization. Our collaboration with their group has led to the development of a supply chain re-engineering tool which is being prototyped at IBM for developing customized applications. In this section, we provide overview of one such application prototyped for asset managers in the IBM supply chain for effective inventory management.

One of the prime concerns while managing a large supply chain is how to control the inventory within the supply chain while providing the required service to customers. It is impossible to have tractable analytical models for these problems under realistic assumptions. In addition, one might be interested in evaluating alterations to the supply chain in various ways (like introducing a new supplier, reducing process lead times) in order to improve the performance. Simulation along with approximate analytical solutions is utilized in the industry to analyze such problems. An ability to make modification to the operational parameters and the structure of the supply chain and evaluate the effect of these modifications is extremely useful in effectively managing the supply chain in a fast changing environment. One such application has been prototyped at IBM for inventory control within the supply chain corresponding to a primary product line. The supply chain under consideration had 11 different types of end products, 1200 different parts in the bill of material and 2000 inventory locations including both IBM internal divisions as well as external suppliers located worldwide. The application will be used to address wide range

of issues, including determining the optimal target inventory levels throughout the network, the effects of customer service, supplier performance, demand variability, and parts commonality among others on the inventory capital (asset) in the supply chain.

The data for this application was collected from several plants that were involved in this business and was formatted to be read in directly by the application. The IBM asset managers specify the supply chain under consideration by instantiating the different manufacturing plants, distribution centers, suppliers and transportation entities involved. The data corresponding to each entity such as products assembled, bill of material associated with products, lead time to produce, transportation delays, holding costs for inventory and transportation cost is read into the application automatically from formatted files once all the entities in the supply chain are connected. This mode of automatic loading of the data was preferred because as the supply chain gets larger, it is difficult to populate each and every entity with data. Based on historical data and future scenarios, the asset manager chooses the likely demand distribution for the supply chain and also feeds in the customer service level that is expected out of the system. Once this is done, an optimization routine is run on the network to decide on the inventory levels to be maintained at various locations within the supply chain. This optimization is based on probabilistic analysis of stock-outs within the supply chain and involves certain simplified assumptions which are explained in greater details in Ettl *et al.* (1996). The value of inventory levels generated by the optimization routine is automatically loaded back into the application. Repeated simulations are conducted and the performance of the system is evaluated in terms of inventory costs and customer service.

During these simulations, the asset manager inputs realistic inventory policies to simulate (which may be different from those assumed in the optimization routine) and may also make modifications to the inventory levels before simulation. One of the reasons for utilizing the optimization routine is to give the asset managers an initial value for inventory levels which is reasonable from an optimization point of view. The application provides an ability to model and simulate, policies and environments that are more realistic than the assumptions used in the optimization routine. As a result, the asset manager can - (1) evaluate the performance of the inventory levels suggested by the optimization routine under a more realistic environment; (2) change parameters such as inventory levels, lead time and transportation time at different locations to better understand the dynamics of the supply chain; (3) make modifications based on his or her experience and evaluate their

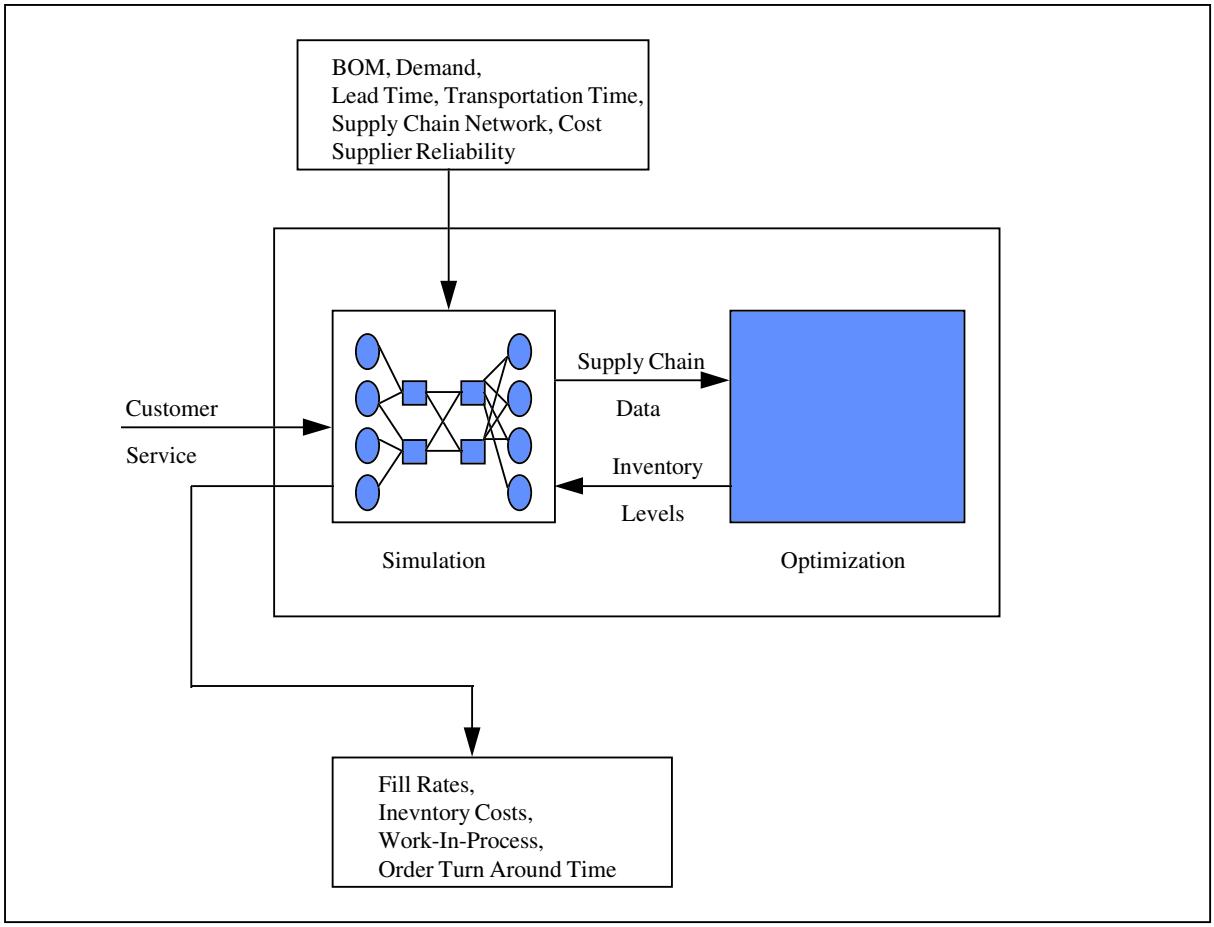

Figure 6: Inventory Servicability Application

performance.

The configuration of the supply chain can be modified by adding new entities or changing production within the supply chain. Evaluation of alternative configurations provides the manager with insights on how changing the supply chain might affect the performance in terms of costs and service. The ability to fine tune the system and evaluate performance under different scenarios makes this application useful for evaluating short term (like setting inventory levels, changing inventory control policies) as well as long-term (like changing a supplier, adding a distribution center) re-engineering efforts. Primitives from our framework reduced the development time for a model using this application significantly. The development time for this application was determined mainly by the time it required to develop the optimization routines and collect data from various plants. This application is currently being introduced into the IBM supply chain for effective inventory management.

7 Conclusions

As manufacturers attempt to increase supply chain performance, there is a critical need to gain a deeper understanding of impact of decisions on their operations as well as those of their partners. Simulation has been found to be one of the popular and suitable mechanisms for understanding supply chain dynamics. Many times supply chain re-engineering decisions are made with a probabilistic view of the future. As a result, there is a necessity for decision support tools that can help managers to understand the costs, benefits and risks associated with various alternatives. In this paper, we have described a simulation based framework for developing customized supply chain models from a library of software components. These components capture generic supply chain processes and concepts, thereby promoting modular construction and reuse of models for wide range of applications. Using these components it is possible to incorporate supply, process and demand uncertainty as well as integrate analytic and heuristic decision procedures. Our approach underscores the importance of models in which different entities in the supply chain operate subject to their own local constraints and objectives, and have different local views of the world. This multi-agent approach enables performance to be analyzed from a variety of organizational perspectives. As evidence of practical utility, a subset of concepts from this framework is being utilized by IBM for supply chain re-engineering efforts.

Several aspects of the framework warrant further investigation. Our current research

directions include - (1) development of features in messages related to cash flows to enable simulation of global environments including currency exchange rates; (2) development of processes to simulate continuous manufacturing; and (3) incorporation of more adaptive agents that are capable of modifying their control policies during simulation based on evolving circumstances.

Acknowledgments

The research was supported by IBM Cooperative Graduate Fellowship and Advanced Research Projects Agency under contracts #F30602-91-F-0016 and #F30602-90- C-0119. The authors wish to thank the associate editor and two anonymous referees whose comments have greatly improved this paper. The authors also wish to thank Dr. Chae An, Dr. Steve Buckley and the business modeling group at IBM T.J. Watson Research Center for introducing the first author to a number of issues in this domain. Our thanks to Dr. Markus Ettl, Dr. Gerry Feigin, Dr. Grace Lin and Prof. David Yao who primarily developed the tool for IBM asset managers.

References

[1] Arntzen, B.C., G.G. Brown, T.P. Harrison and L.L. Trafton, Global Supply Chain Management at Digital Equipment Corporation, *Interfaces*, 25(1), 69-93, 1995.

[2] Bond A.H. and Les Gasser, An analysis of problems and research in DAI, *Readings in Distributed Artificial Intelligence*, Morgan Kaufman Publishers, Inc., San Mateo, California, 1988.

[3] Billington C., Strategic Supply Chain Management, *OR/MS Today*, 20-27, April 1994.

[4] Cash J.I. and B.R. Konsynski, IS redraws competitive boundaries, *Harvard Business Review*, 63(2), 134-42, 1985.

[5] Clark A.J., An informal survey of multiechelon inventory, *Naval Res. Logist. Quart.*, 19, 621-650, 1972.

[6] Clark A.J. and H. Scarf, Optimal policies for a multiechelon inventory problem, *Management Science*, 6, 475-490, 1958.

[7] Cohen M.A. and H.L. Lee, Strategic Analysis of Integrated Production-Distribution Systems: Models and Methods, *Journal of Operations Research*, 36(2), 216-228, 1988.

[8] Cohen M.A. and S. Moon, Impact of Production Scale Economies, Manufacturing Complexity and Transportation Costs on Supply Chain Facility Networks, *Journal of Man-*

ufacturing and Operations Management, 3, 269-92, 1990.

[9] Efficient Consumer Response Report (ECR), *Enhancing Consumer Value in the Grocery Industry*, Kurt Solomon Associates Inc., Food Marketing Institute, 1993.

[10] Ettl M., G.E. Feigin, G.Y. Lin and D.D. Yao, A Supply Network Model with Base-Stock Control and Service Requirements, IBM Research Report, RC20473, June, 1996.

[10] Feigin G., C. An, D. Connors and I. Crawford, Shape Up, Ship Out, *OR/MS Today*, 24-30, April 1996.

[11] Hall R., *Zero Inventories*, Dow-Jones Irwin, Homewood IL, 1983.

[12] Hayes R., S. Wheelright and K.B. Clark, *Dynamic Manufacturing*, The Free Press, New York, 1988.

[13] Helper S., How much has really changed between US manufacturers and their suppliers ?, *Sloan Management Review*, 32(4), 15-28, 1991.

[14] Henkoff R., Delivering the goods, *Fortune*, vol. 140, no. 11, Nov 28, 1994, E34-47.

[15] Kumar A., P.S. Ow and M.J. Prietula, Organizational simulation and information systems design: An operations level example, *Management Science*, 39(2), 218-40, 1993.

[16] Kwok A. and D. Norrie, Intelligent agent systems for manufacturing applications, *Journal of Intelligent Manufacturing*, 4(4), 285-93, 1993.

[17] Lee H.L. and C. Billington, Managing Supply Chain Inventory : Pitfalls and Opportunities, *Sloan Management Review*, 33(3), 65-73, 1992.

[18] Lee H.L. and C. Billington, Material Management in Decentralized Supply Chains, *Journal of the Operation Research*, 41(5), 835-847, 1993.

[19] Lyons.T.F., A.R. Krachenberg, and J.W. Henke Jr., Mixed motive marriages : What's next for buyer-supplier relations ?, *Sloan Management Review*, 1990.

[20] Malone T., Modeling coordination in organizations and markets, *Management Science*, 33(10), 1317-1332, 1987.

[21] Newhart D.D., K.L. Scott and F.J. Vasko, Consolidating Product Sizes to Minimize Inventory Levels for a Multi-Stage Production and Distribution Systems, *Journal of the Operational Research Society*, 44(7), 637-44, 1993.

[22] Pan J.Y.C., J.M. Tanenbaum and J. Glicksman, A Framework for Knowledge-Based Computer Integrated Manufacturing, *IEEE Transactions on Semiconductor Manufacturing*, 2(2), 87-100, 1989.

[23] Pyke D.F., and M.A. Cohen, Performance Characteristics of Stochastic Integrated Production-Distribution Systems, *European Journal of Operations Research*, 68(1), 23-48,

1993.

[24] Pyke D.F., and M.A. Cohen, Multiproduct Integrated Production Distribution Systems, *European Journal of Operations Research*, 74(1), 18-49, 1994.

[25] Roboam M., K. Sycara and M.S. Fox, Organization modeling as a platform for multi-agent manufacturing systems, *Proceedings of Computer Applications in Production and Engineering*, Bordeaux, France, 1991.

[26] Sabel C.F., H. Kern and G. Herrigel, Collaborative Manufacturing: New Supplier Relations in the Automobile Industry and the Redefinition of the Industrial Corporation, *IMVP International Policy Forum*, MIT, May 1989.

[27] Sadeh N., Micro-opportunistic Scheduling: The MICRO-BOSS Factory Scheduler, *Technical Report*, The Robotics Institute, Carnegie Mellon University, 1994.

[28] Smith S.F., The OPIS Framework for Modelling Manufacturing Systems, *Technical Report*, The Robotics Institute, Carnegie Mellon University, 1989.

[29] Srinivasan K., S. Kekre and T. Mukhopadhyay, Impact of Electronic Data Interchange Technology on JIT Shipments, *Management Science*, 40(10), 1291-1305, 1994.

[30] Svoronos.A and P. Zipkin, Evaluation of one-for-one replenishment policies for multiechelon inventory systems, *Management Science*, 37(1), 68-83, 1991.

[31] Swain J.J., Simulation Survey: Tools for Process Understanding and Improvement, *ORMS Today*, 64-75, August 1995.

[32] Swaminathan J.M., Supply Chain Library, *IBM Internal Report*, Manufacturing Research, 1994.

[33] Swaminathan J.M., Quantitative Analysis of Emerging Practices in Supply Chains, *Unpublished Ph.D. Dissertation*, GSIA, Carnegie Mellon University, 1996.

[34] Swaminathan J.M., N.M. Sadeh and S.F. Smith, Information Exchange in Supply Chains, *Technical Report*, The Robotics Institute, Carnegie Mellon University, 1995.

[35] Towill D.R., M.M. Naim and J. Wikner, Industrial Dynamics Simulation Models in the Design of Supply Chains, *International Journal of Physical Distribution and Logistics Management*, 22(5), 3-13, 1992.

[36] Tzafestas S. and G. Kapsiotis, Coordinated Control of Manufacturing/Supply Chains using Multi-Level Techniques, *Computer Integrated Manufacturing Systems*, 7(3), 206-212, 1994.

[37] Womack J.P., D.T. Jones, and D. Roos, *The Machine that Changed the World*, Rawson Associates, 1990.