# A hierarchical approach to supply chain simulation modeling using the Supply Chain Operations Reference model

**Guruprasad Pundoor***
Operations Research Analyst
Federal Express
Memphis, TN 38125
Email: gpundoor@fedex.com
*Corresponding author

**Jeffrey W. Herrmann**
Department of Mechanical Engineering and Institute for Systems Research
University of Maryland
College Park, MD 20742, USA
Email: jwh2@umd.edu

**Abstract:** Simulation is a very useful tool for predicting supply chain performance. Because there are no standard simulation elements that accurately represent the activities in a supply chain, there exist a variety of approaches for developing supply chain simulation models. This paper is an attempt to improve this situation by describing a novel supply chain simulation framework that follows the Supply Chain Operations Reference (SCOR) model. This framework has been used for building powerful simulation models that integrate discrete event simulation and spreadsheets. The simulation models are hierarchical and use submodels that capture activities specific to supply chains. The SCOR framework provides a basis for defining the level of detail in such a way that it includes as many features as possible, while not being industry specific. This approach enables the reuse of submodels, which reduces the model development time. The paper describes the implementation of the simulation models and details how the submodels interact with each other. The paper also explains a study that used this framework to analyse the impact of rescheduling frequency on the supply chain performance.

**Keywords:** supply chains, simulation models, Supply Chain Operations Reference (SCOR) model

**Biographical notes:** *Guruprasad Pundoor* works as an operations research analyst with the Federal Express. He has a Ph.D. in Management Science from the Robert H. Smith School of Business, University of Maryland, College Park. He also holds a M.S. in Systems Engineering from the University of Maryland, College Park, and a B.Tech in Mechanical Engineering from the Indian Institute of Technology, Madras, India. His current research interests include applying optimisation techniques in the areas of supply chain scheduling and revenue management.

*Jeffrey W. Herrmann* is an associate professor at the University of Maryland, where he holds a joint appointment with the Department of Mechanical Engineering and the Institute for Systems Research. He is the director of the Computer Integrated Manufacturing Laboratory. He is a member of INFORMS, ASME, IIE, and SME. He is the faculty advisor for the University of Maryland ASME Student Section. Dr. Herrmann earned his B.S. in applied mathematics from Georgia Institute of Technology. As a National Science Foundation Graduate Research Fellow from 1990 to 1993, he received his Ph.D. in industrial and systems engineering from the University of Florida. His publications cover topics in process planning, production scheduling, manufacturability evaluation, and manufacturing facility design. His current research interests include the design and control of manufacturing systems, the integration of product design and manufacturing system design, and decision-making systems in product development. For more details, please visit: http://www.isr.umd.edu/~jwh2/jwh2.html

## 1 INTRODUCTION

A supply chain is a network of suppliers, manufacturers, distributors, and retailers who are collectively concerned with the conversion of raw materials into goods that can be delivered to the customer. Three kinds of flows are to be considered in any supply chain: material flow, information flow, and cash flow. Material flows from suppliers and manufacturers to distributors and retailers and, finally, to customers. This flow refers to the transportation of products from one participant to another on one hand and to the movement of raw materials and parts within the shop floor on the other. Information flow refers to the data that are generated every time a change in the system status occurs. For example, every customer order generates information that is used to fulfil the order. Cash flow is the flow of money in the supply chain from customers to retailers and back to suppliers. For simulating a supply chain, it is important to model the interaction between various participants accurately, in addition to considering these three kinds of flows.

A supply chain is a dynamic, stochastic, and complex system. The performance of any particular participant in a supply chain depends to a large extent on the behaviour of other participants. Optimising the performance of each participant is important, but for improving the overall performance of a supply chain, it is necessary to view the system as a whole. This makes supply chain management very complicated.

Supply chain simulation models can be used to improve supply chain decision-making. The relevant decisions can be classified into three categories (Gaither and Frazier, 2002): strategic, operating, and control. Strategic decisions such as selecting the location of a facility have long-term significance. Operating decisions refer to decisions about production to meet demand. These decisions are made on a weekly or a monthly time frame. Control decisions are concerned with problems in execution. This can be classified as disruption management. Examples include the decisions to be taken when a certain machine in the shop floor fails. Simulation models can be used to evaluate policies (such as inventory management policies) or to predict the outcome of a specific alternative.

Each participant of the supply chain performs a distinct set of activities. Despite differences between these sets, a number of processes are common to the participants of the supply chain. Components that represent these common process elements can be used to construct a model of the entire supply chain. This enables the principle of reuse in the bottom-up development of a model. Independent components with well-defined interfaces also promote reusability.

In addition to this, a variety of approaches for building supply chain simulation models have been described. Some of these approaches used general-purpose discrete event simulation, others developed specialized software, and still others used distributed simulation.

Jain *et al.* (2001) observe that the level of detail included in the development of a simulation model should be appropriate to the objective of the study. The paper describes a high-level supply chain simulation model that includes order fulfilment, procurement, forecasting, and replenishment. Their approach uses general-purpose simulation software because it lets the user select the desired level of abstraction.

Bhaskaran (1998) used an automobile supply chain simulation software originally developed to GM's specifications. Chatfield *et al.* (2001) describe an approach that automatically generates supply chain simulation models. The analyst must first describe the supply chain structure using a special modelling language. A model generation routine creates a simulation model using a library of Java classes.

Eliter *et al.* (1998) worked on the concept of Agent Programs. An agent consists of a body of software code that supports a well-defined application programmer interface and a semantic wrapper that contains a wealth of information. As part of the work, the team developed agents for various functions of supply chain management systems. A simulation model of a supply chain application based on agents was built using commercial software such as Microsoft Access and ESRI's MapObject. Swaminathan *et al.* (1998) describe a supply chain modelling framework through software components for representing various types of supply chain agents such as retailers, manufacturers, and transporters.

Commercial vendors also offer supply chain simulators such as IBM Supply Chain Simulator (Bagchi *et al.*, 1998), Supply Chain Builder (Simulation Dynamics, 2001), and e-SCOR (GenSym, 2003).

Although these supply chain simulation models are discrete-event simulation models, some supply chain variables such as inventory levels can be viewed as continuous variables. (See, for example, Lee *et al.* (2002).) Our approach includes planning activities that manage these variables using Excel VBA (see Section 3.1).

The proliferation of supply chain simulation models has yielded competing approaches. Because there are no standard elements that represent accurately the activities in a supply chain, there exist a variety of approaches for describing supply chain models. To resolve this problem, this research adopted the Supply Chain Operations Reference (SCOR) model (Supply-Chain Council, 2004), which has been proposed as a standard for describing supply chain management processes, their relationships, and best practices. According to the Supply Chain Council, SCOR is a process reference model designed for effective communication among supply-chain partners, and is used to describe, measure, and evaluate supply-chain configurations (Supply-Chain Council, 2004). Barnett and Miller (2000) describe specialized supply chain simulation software that implements SCOR.

Our goal is to implement supply chain simulation models using the SCOR model with reusable components from general-purpose discrete-event simulation software to facilitate model construction. Thus, firms that are using the SCOR model will be able to create simulation models of their supply chains more easily. With the current tools, it is very difficult to build generic supply chain simulation models. One of the reasons is the lack of standardization, while another is the effort involved in modelling the large number of activities involved. As for using off-the-shelf discrete event simulation packages directly, they do not provide custom made modules specific to supply chain simulation. The basic modules in these packages have to be combined to represent various supply chain activities. This is a very time consuming task. Moreover, when reusable supply chain simulation modules are not available, it becomes difficult for the analyst to modify the structure of the supply chain in order to evaluate possible alternatives.

The simulation modules that we have defined do not cover all possible operational details. While, in many cases, the details in the modules are sufficient to describe the supply chain activities, some users may need to modify some modules to model their supply chain more accurately, depending on the specifics of the activities involved. The simulation modules are easy to modify, and since each module has a well-defined set of interfaces with the other modules, the other modules can be used as-is. We have constructed a variety of simulation models using the framework proposed in this paper. These are described in more detail in Pundoor (2002). In addition, a more detailed description of the model and links to examples are available online at the following URL: http://www.isr.umd.edu/Labs/CIM/SC_Simulation/

The remainder of the paper is organized as follows. Section 2 presents our new framework for building supply chain simulation models. Section 3 describes the implementation using Arena and Microsoft Excel, explains how the submodels interact, and describes cash flow and the performance measures. Section 4 deals with a sample implementation of the SCOR supply chain simulation framework. Section 5 concludes the paper.

## 2 SUPPLY CHAIN SIMULATION FRAMEWORK

The primary objective of this work is to build supply chain simulation modules that are *reusable*. To achieve this, our approach emphasizes on standardized modules. The modelling approach is also hierarchical. (Note, however, that the hierarchical modelling approach does not presume or require a hierarchical decision-making structure within the supply chain.) This approach lets the analyst capture the system at different levels of detail, in addition to facilitating a bottom-up development approach. Using these modules, supply chain simulation models with great flexibility can be built with very little effort. The hierarchical simulation modelling approach presented here is based on the Supply Chain Operations Reference model, Version 4.0, proposed by the Supply Chain Council (2000). The SCOR model was developed to describe the business activities associated with all phases of satisfying a customer's demand. By describing supply chains using SCOR process building blocks, the model can be used to describe supply chains that are very simple or very complex using a common set of definitions. SCOR is founded on four distinct supply chain management processes: *Plan, Source, Make,* and *Deliver*. Supply chains can be described using these building blocks. (Later versions of SCOR include a *Return* process as well.)

The SCOR model also distinguishes between Planning, Execution, and Enable level process types. Planning processes balance aggregate demand across a consistent planning horizon. Planning processes generally occur at regular intervals. Execution processes are triggered by planned or actual demand that changes the state of products. These include scheduling and sequencing, transforming materials and services, and moving product. Enable processes prepare, maintain and manage information or relationships upon which planning and execution processes rely. Combinations of a SCOR process and a process type form *Process Categories*. Each of the *Process Categories* consists of *Process Elements*. The simulation modules that we have defined correspond to *Process Elements*. Thus, they can be used for any type of supply chain configuration.

For explaining our approach, it is convenient to identify three kinds of participants: consumers, producers, and traders. Consumers are those participants who place orders for finished products, but do not supply any products to any other participants. They are the most downstream participants in the supply chain. Producers are the most upstream participants. Producers supply parts to other participants, but do not receive any. Traders are the intermediate participants in the supply chain. Traders both place orders with some participants and deliver orders to other participants. Traders include manufacturers, distributors, and retailers.

In this framework, a simulation model of a supply chain has three levels. The first level is the simulation model. The second level has submodels that correspond to the supply chain participants (consumers, producers, and traders). The third level has submodels that correspond to the process elements (across all process categories) that each participant performs. Figure 1 displays the corresponding hierarchy of submodels. Each participant submodel includes a subset of the process element submodels shown in Figure 1.

There are small differences in the submodels for consumers, traders, and producers. In the case of the producers, raw material sourcing is not performed. A sufficient amount of raw materials is assumed to be available all the time. (This can be easily modified to represent a different situation such as a raw material inventory position based on the consumption rate and a sourcing policy.) The consumer acts as a place for receiving the products corresponding to the orders that he places. So the consumer does not perform production and delivery activities. Because participants such as distributors or retailers do not have any manufacturing processes, the corresponding participants do not have produce and test submodels.

Each process element is implemented as a separate submodel that represents a specific activity in the supply chain. Each process element submodel has clearly defined interfaces, which are used to integrate the submodels. The participant submodels contain process element submodels and other submodels needed to initialise the simulation model. One of the important decisions to make while developing the model is the level of detail at which to implement it. For example, a manufacturer can be modelled at a very detailed level, representing the operation of each machine in the factory and the flow of raw material on the shop floor. On the other hand, it can also be modelled at a very high abstract level, with an approximate measure of factory-wide parameters such as production lead-time. A detailed model requires more effort to construct and is more difficult to modify. On the other hand, an abstract model may fail to capture some important information. Our simulation framework attempts to balance these by incorporating many of the parameters required to represent each activity accurately while avoiding too much detail. Also, the hierarchical nature of the modules provides the user with the flexibility to implement a part of the model at a greater level of detail if necessary.

## 3 IMPLEMENTATION

This section describes the implementation of the framework using Arena and Microsoft Excel, the initialisation of a supply chain simulation model, the interactions of the process element submodels, and the cash flow and performance measures. We have made available a model that implements this framework. Sample files for this demo model can be downloaded from the following link: http://www.isr.umd.edu/Labs/CIM/SC_Simulation/Demo_Model/. The user should have Microsoft Excel and the professional version of Arena 5.0 or higher installed in order to run this simulation model.

### 3.1 Simulation and Spreadsheet Integration

The simulation models were built using Arena 5.0 and Microsoft Excel 2000. The Arena software interacts with Microsoft Excel using Arena VBA (Visual Basic for Applications), as shown in Figure 2. VBA provides the methods and routines for applications to interact with each other. Each participant in the supply chain has its own set of modules. For building the supply chain simulation model, these modules are put together and connected using standard interfaces that represent the material, information, and cost flow. Each participant of the supply chain also has an Excel workbook associated with it. The Arena submodels associated with a participant include the VBA blocks that communicate with tables in the corresponding Excel workbook (to get or save data) using Excel VBA. When invoked, an Arena VBA block accesses the corresponding Excel file and reads or writes data using Excel VBA.

Planning activities are carried out using Excel VBA. Execution is carried out in Arena. Enable processes are modelled as input to the simulation either in the form of Excel data or as parameters in the Arena model. Arena triggers various planning activities in Excel at periodic intervals (e.g. checking inventory) or based on random events (e.g. customer placing an order). Each planning activity checks the system status and takes actions depending on the status. The Excel workbooks record the status of the system and evaluate the performance measures. Periodic clean up actions prevent the Excel files from becoming too large in the course of a simulation run. The customer orders and purchasing orders that have been filled are archived once the performance measures relating to those orders are recorded. The archived customer orders remain in a text file that can be viewed at the end of the simulation run if desired.

### 3.2. Model Initialisation

Constructing the supply chain simulation model requires constructing the Arena submodels and the Excel workbooks for each participant, since both the Arena submodels and the Excel workbooks include data needed to specify the complete model. While some of these data are dependent on the modules (e.g., processing time at a server), some others are dependent on the products (e.g., bill of materials).

### 3.3. Model Execution

This section describes how the submodels work together to execute the key activities that occur in supply chain operations. Model execution is determined by the information flow within and between the various participants. The information flow consists of two types: the first type records the status of the system, and the second type triggers events in the model. The simulation progresses due to these events. At a high level, the following are the activities that trigger actions in Excel and Arena and determine the course of the simulation:

1. A trader checks the inventory and places orders for raw materials with other traders or producers if necessary. (Orders placed by consumers are a special case of this activity.)

2. A trader or producer checks the existing open orders for production and obtains the production plan based on material availability.
3. A trader or producer checks the open orders for delivery to construct a delivery plan.

The following subsections explain each of these activities.

### 3.3.1 Sourcing

Traders perform sourcing at periodic intervals. The trader orders raw materials from his supplier based on an inventory control policy, which, in the models that we have created, is a periodic (*R, s, S*) policy. (The user has the flexibility to change this to any other inventory policy.) *R* is the interval at which inventory is checked, *s* is the reorder level, and *S* is the order up to quantity. These values are defined for each type of product and are specified in the *Inventory Management* table in the corresponding Excel workbook. The net inventory position is calculated using the on-hand inventory, the on-order inventory, the allocated inventory, and the backorders.

For each trader, the *Schedule Product Deliveries* submodel in Arena, which corresponds to module S2.1 in SCOR, periodically triggers an event that invokes the Excel procedure for checking the inventory levels (Figure 3). The values for on hand inventory, inventory on order, allocated inventory, and backorders for each component can be obtained from the *Item Master* table. Excel VBA calculates the sourcing quantity based on these values. The trader's (or producer's) name for each component is obtained from the *Item Master* table.

Figure 4 shows the process elements and worksheets involved in the receipt of sourced products. Sourced products are received in three stages: *Receive Product*, *Verify Product*, and *Transfer Product*. In the *Receive Product* stage, the sourced products seize a resource at the receive module. The processing time distribution depends upon the product type. After receiving, the product goes through the *Verify Product* stage, which delays the movement of the sourced product. The *Transfer Product* stage seizes a resource that moves the verified products into the raw material inventory. Each of these processes has an associated cost and this cost is added to the sourced product depending on the amount of time the product spends at each resource.

Once the sourced products reach the raw material inventory (after the *Transfer Product* stage), the Arena VBA block calls the Excel VBA procedure for updating the inventory status in the *Item Master* table. It also updates the *Purchase Action Report* and the *Material Release* table, which tracks the values of the raw material available in the inventory.

### 3.3.2. Checking open orders for production

All unfinished customer orders have their status indicated by a tag in the *Customer Order* table. The status of an existing order is either *Received, In process, FGI, In transit,* or *Delivered*. The *Customer Order Tracking* table lists the orders that are open for production (their status is *Received*). That is, these orders have been received, but they are not yet scheduled for production. Periodically, these orders are checked for production release. The interval between each such check depends on the production rescheduling period. If material for processing the whole order is available, then it is released for production. All of the open orders for which material is available are released at the same time for production. If the available material is insufficient, the order remains open and is checked again during the next production order release cycle.

Figure 5 shows the process elements and Microsoft Excel worksheets that are used for simulating the production activity. During each production order release cycle (the *Schedule Production* stage), Excel VBA checks the inventory status in the *Item Master* worksheet to identify orders that can be released for production. In a more detailed set-up, algorithms could be implemented that reject an existing order based on some criterion, or perform some other complicated production scheduling operations. This can be achieved by modifying the appropriate simulation modules while keeping their interfaces with the other modules intact. For checking the material availability, both the order size and the bill of materials for the corresponding product have to be considered. This is carried out in Excel VBA. During each planning cycle, open orders are listed for processing based on a heuristic. Raw material requirements are calculated using the bill of materials. Whenever an order is released for production, the necessary raw material is removed from the inventory and its status is changed from *Received* to *In process.*

The released orders seize the *Issue Product* resource, which transports the raw materials from the raw material inventory to the shop floor. The processing time distribution for this stage depends upon the product and the order quantity.

After the raw material has been issued to the shop floor, the order goes through the *Produce and Test* stage. (This stage is absent in participants such as the distributors and the retailers that do not perform any production activities.) The test stage includes a rework loop that sends a portion of the orders for rework.

The order then enters the *Package* stage. After packaging, the order moves to the *Stage Product* stage, and then the order is ready for delivery and moves to the finished goods inventory. (Note that this stage uses the same *Customer Order* table that the *Schedule Production* stage uses.) At this time, the status of the order changes from *In process* to *FGI*. The order waits in the finished goods inventory until a delivery plan releases it for delivery. Each of the processes mentioned above has costs associated with it and the costs are added to the order using job order costing method.

### 3.3.3 Checking completed orders for delivery

The finished goods inventory status is checked periodically. The *Customer Order Tracking* table keeps track of the orders that are available for delivery in the finished goods inventory. As shown in Figure 6, these orders are sent for delivery during the *Schedule Delivery* process. The delivery process requires seizing a transporter resource. The processing time here corresponds to the transportation time from the producer (or trader) to the customer. Each order is delivered separately. The cost for transportation gets added to the cost of the order. Once the order is delivered, its status is changed to *Delivered.* The price for the order is obtained from the *Item Master* table. This value, along with the accumulated cost, is used for calculating the profit. After the performance measures corresponding to the order have been recorded, it is removed from the Excel file and archived in a text file.

### 3.4. Cash Flow

In addition to time based performance measures (discussed in the next section), the simulation model also records financial performance measures. Cash flow is obtained by associating costs to each order. Cost accumulation methods (the manner in which costs are collected and identified with specific customers, jobs, batches, orders, departments and processes) vary from firm to firm. The modules provide flexibility on the cash flow techniques and detailed accounting methods can be implemented in the supply chain as long as they do not modify the interface between the simulation modules. In the models developed, job order costing method is followed. In job order costing, costs are accumulated by jobs, orders, contracts or lots. In the simulation model, each order is considered as a job and costs are assigned to it. Direct material, direct labour, and overhead rates are considered for assigning costs to each order. All the process costs, including the manufacturing costs, are applied to orders using predetermined rates along with an overhead rate associated with each activity. Direct material cost is obtained during order release using the first-in-first-out policy for the raw material inventory. Inventory holding costs are calculated at all the stages. The cost assigned to an order at a particular resource depends on the amount of time the resource was utilized by the order. Costs at various stages are added to arrive at the final cost for the order.

### 3.5. Performance Measures

Periodically, Arena VBA triggers Excel procedures that calculate the performance measures based on the entries in the corresponding Excel sheets. At the end of each replication, these performance measures are put together and the overall performance measures for the entire replication are calculated. The performance measures include cycle time, percent tardiness, inventory, cost performance, and resource utilization. Order based performance measures are calculated based on the orders that have been delivered during any given period. For purposes of cycle time calculations, the whole process from placing of an order to the delivery of the finished product at the customer site is divided into four stages: order receipt to start build, start build to finished goods inventory, finished goods inventory to release for delivery, and release for delivery to delivery at customer site. The cycle time refers to the average time at each of the stages, the average being taken over the customer orders. The overall cycle time is calculated as the average time between the placing of an order by the customer and the delivery of that order by the producer (or trader) at the customer site. Each product has an associated lead-time. Whenever an order is placed, its estimated delivery date is given based on the lead-time for that product. If the order is delayed beyond its estimated delivery date, then the order is considered tardy. The percentage of orders that were delivered after the due date is calculated as the percentage tardy performance measure. For calculating the resource utilization, variables are used to keep track of the amount of time the resource was busy in any given period. Cost performance measures are calculated based on job order costing. Costs are associated with each order and these values are used to obtain performance measures such as cost of goods sold.

*Delivery Performance:* Delivery performance includes the average cycle time at each stage, the overall cycle time, and the percentage of orders that were tardy. For calculating the cycle times, the four stages mentioned above are considered: order receipt to start build, start build to finished goods inventory, finished goods inventory to release for delivery, and release for delivery to delivery at customer site. The sum of the average cycle times at these four stages gives the overall cycle time.

*Inventory Performance:* Inventory is measured in dollars. Each inventory performance measure is the average of the inventory at the beginning of the period and the inventory at the end of the period. The inventory performance measures include raw material, work in process, and finished goods inventory.

*Inventory Holding Expenses:* Each product has an inventory holding cost associated with it. Inventory holding expenses are calculated based on the average inventory level.

*Inventory Days of Supply:* This is calculated based on the cost of goods manufactured and the average inventory level. This ratio measures the number of days it takes to sell the entire stock of inventory.

*Cost of Goods Sold:* The cost of goods sold is calculated based on the production costs, purchases, work in process, and finished goods inventory. For a manufacturing firm, cost of goods sold is the manufacturing expenses, along with other expenses for goods sold during the period, including raw material, direct labour, and overhead. For a retail firm, the manufacturing process is not present. Cost of goods sold can be used to find the gross profit during the period. The gross profit is defined as the difference between the sales and the cost of goods sold. The total sales can be obtained from the total price for the orders delivered during the period.

*Cost of Goods Manufactured:* Cost of goods manufactured is the cost of orders that were put in the finished goods inventory during the period. This includes the cost of orders that were released for production in an earlier period but completed during the current period. This value is dependent on the manufacturing expenses for the period, including the overhead, and the work in process inventory at the beginning and end of the period.

*Process Element Utilization:* Process element utilization for each of the resources is calculated at the end of the period. This value is dependent on the time for which the corresponding resources were busy during the period.

## 4 USING THE SCOR SIMULATION FRAMEWORK: A STUDY ON RESCHEDULING

In this section, we give a brief overview of a study on the impact of rescheduling frequency on the performance of supply chains. The supply chain simulation models used in the study were implemented using the simulation framework and modules discussed in this paper. The objective of this section is to show a typical example of how the simulation framework could be used to make managerial decisions.

Many supply chain activities involve uncertainty. It is important to keep track of the system status on a regular basis in order to obtain desired performance. Variability along with limited resources makes it difficult to manage the system. The system performance depends to a great extent on the rescheduling policies for various planning activities. The rescheduling period for a particular activity defines how frequently the planning for that activity is carried out. When a rescheduling period is short, more effort is spent on planning activities, and plans change frequently, although this might lead to a better control over the system. When a rescheduling period is long, it becomes difficult to respond to unexpected events and manage the system efficiently. In this study, we analyse the direct and indirect effects of varying the rescheduling periods of various activities on the performance of a supply chain. The following activities were considered: order release for sourcing of raw materials from the suppliers, release of customer orders for production, and release of customer orders for delivery. So a rescheduling period of 24 hours for order release for sourcing means that the order release process for sourcing occurs once every day.

We consider two different supply chains. The first supply chain has five participants: two suppliers, one manufacturer, and two customers. The manufacturer produces two kinds of products: Product 1 and Product 2. Each customer places orders for both the products. Each unit of Product 1 consists of one unit of Component 1 and one unit of Component 2. Each unit of Product 2 consists of one unit of Component 1 and one unit of Component 3. Supplier 1 supplies Component 1 and Component 2, while Supplier 2 supplies Component 3. Figure 7 shows the organizations in the supply chain and their relationships in the flow of material. (Note that this figure does not describe management processes and is not covered by the SCOR model.) After modelling each of the organizations using the SCOR processes, process categories, and process elements, we created a simulation model using the simulation modules for each process element.

The second supply chain has ten participants: four suppliers, one manufacturer, one distributor, two retailers, and two customers. The manufacturer produces three kinds of products: Product 1, Product 2, and Product 3. Each customer places orders for all the products with the retailer. Each unit of Product 1 consists of one unit of Component 1 and one unit of Component 2. Each unit of Product 2 consists of one unit of Component 1 and one unit of Component 3. Each unit of Product 3 consists of one unit of Component 1 and one unit of Component 4. Supplier 1 supplies Component 1, Supplier 2 supplies Component 2, Supplier 3 supplies Component 3, and Supplier 4 supplies Component 4. Figure 8 shows the organizations in this supply chain and their relationships. We created a simulation model of this supply chain as we did for the one described above.

Due to space restrictions, we refrain from giving a detailed explanation of each supply chain and their various parameters. A complete description of the supply chains is available online (Pundoor, 2002, http://www.isr.umd.edu/Labs/CIM/SC_Simulation/). In the simulation runs, we vary the rescheduling periods from very short (close to continuous review) to very long (infrequent reviews). We also compare cases with synchronization between participants to those cases where there is no synchronization. For example, if there is synchronization between a supplier and a manufacturer, the supplier would carry out the rescheduling activities only after orders from the manufacturer have been received. On the other hand, if there is no synchronization, the supplier would carry out rescheduling activities without waiting for the manufacturer's orders and hence the manufacturer's sourcing orders may have to wait for one complete rescheduling period before getting processed. We compare the performance using cycle times and tardiness performance under each scenario.

The results show that the frequency of planning activities is an important factor that needs to be considered while analysing any supply chain. A high rescheduling period at a particular activity may have adverse effect not only on that activity, but also on the performance of downstream activities as well. From the simulation runs, we can conclude that very low rescheduling periods are always good for the system. But we note that the marginal improvements obtained by reducing the rescheduling periods are not significant beyond some point. Also, a small rescheduling period means more effort in planning. Too frequent change of plans will complicate the operations of the firm and the supply chain especially under conditions of stochastic, dynamic demand. This may create "system nervousness" that outweighs any benefits obtained through frequent updating of information. Hence there is a trade-off between the supply chain performance and the effort required. We also observe that synchronization of activities across participants of the supply chain has a significant impact on the performance. We analysed two extreme cases of synchronization. In backward synchronization, whenever an upstream participant makes planning decisions, all the downstream participants would already have carried out their planning activities for that period. In such a scenario, the upstream participant can include all the requirements of the downstream activities without any time delay. Forward synchronization is exactly the opposite. Here, an upstream participant carries out the planning activities before all the downstream participants. The results show that backward synchronization improves the performance of the system significantly.

Backward synchronization requires coordination across the entire supply chain. All the participants should be willing to share information with their upstream participants based on a predetermined schedule. The simulation modules proposed in this paper can be used to quantify the benefits or disadvantages of switching to a different structure of information sharing or a different frequency of rescheduling.

## 5 SUMMARY AND CONCLUSIONS

As companies concentrate on improving the performance of the entire supply chain instead of viewing it as a set of independent organizations, coordination among various organizations becomes important. Simulation is a very effective way of evaluating different scenarios in such an environment. With the advent of more powerful computers, it has become easier to simulate complex systems. But the amount of time needed to develop the simulation model can be quite high. Libraries of reusable submodels can be used to build supply chain models with less time and effort, thus increasing the amount of time available for evaluating the system.

Arena, like other simulation software, offers numerous features to simulate discrete event systems. But the modules available in Arena are at a very basic level compared to those required in supply chain simulation models. Developing hierarchical models with reusable submodels can overcome this limitation. In addition, by using Arena VBA, the simulation model can communicate with other applications such as Microsoft Excel. By combining the simulation capabilities of Arena and the spreadsheet capabilities of Microsoft Excel, we have constructed a very efficient and flexible library for developing supply chain simulation models. In order to make the submodels standardized (and thus more useful), we have followed the Supply Chain Operations Reference model.

**Note:** An abridged version of this paper (Herrmann *et al.*, 2003) was presented at the ASME 2003 International Design Engineering Technical Conferences and Computers and Information In Engineering Conference.

## ACKNOWLEDGEMENTS

This research was supported as part of a collaboration entitled Enhancing Collaborative Research in Industrial and Systems Engineering. The following organizations participated in the collaboration, which was sponsored by the National Science Foundation: Department of Industrial and Systems Engineering, North Carolina A&T State University; The Logistics Institute, Georgia Tech; and the Institute for Systems Research, University of Maryland. The research was conducted in the Computer Integrated Manufacturing Lab. The authors appreciate the help provided by Edward Lin and Paul Stanfield. The authors also thank Aparna Sundaram, Dr. Sanjay Jain, and the two anonymous referees for their helpful comments and suggestions

## REFERENCES

Bagchi, S, Buckley, S. J., Ettl, M., and Lin, G. Y. (1998) `Experience using the IBM supply chain simulator', *Proceedings of the 1998 Winter Simulation Conference*, pp. 1387-1394.

Barnett, M.W., and Miller, C. J. (2000) `Analysis of the virtual enterprise using distributed supply chain modeling and simulation: an application of e-SCOR', *Proceedings of the* 2000 *Winter Simulation Conference*, pp. 352-355.

Bhaskaran, S. (1998) `Simulation analysis of a manufacturing supply chain', *Decision Sciences*, Vol 29(3), pp. 633-657.

Chatfield, D. C., Harrison, T. P., and Hayya, J. C. (2001) `SISCO: a supply chain simulation tool utilizing Silk and XML', *Proceedings of the 2001 Winter Simulation Conference*, pp. 614-622.

Eliter, T., Subrahmanian, V. S., and Pick, G. (1998) `Heterogeneous active agents', *Computer Science Technical Report Series*, University of Maryland, College Park.

GenSym. (2003) `e-SCOR Overview' http://www.gensym.com/ supplychain/escor_overview.shtml.

Gaither, N., and Frazier, G. (2002), *Operations Management*, Southwestern Thomson Learning, Cincinnati, Ohio.

Herrmann, J. W., Lin, E., and Pundoor, G. (2003) `Supply chain simulation modeling using the Supply Chain Operations Reference model', *Proceedings of the ASME 2003 International Design Engineering Technical Conferences and Computers and Information In Engineering Conference*, DETC2003/CIE-48220.

Jain, S., Workman, R. W., Collins, L. M., and Ervin, E. C. (2001) `Development of a high-level supply chain simulation model' *Proceedings of the 2001 Winter Simulation Conference*, pp. 1129-1137.

Kelton, W. D., Sadowski, R. P., and Sturrock, D. T. (1998), *Simulation with Arena*, McGraw-Hill.

Lee, Y. H., Min, K. C., Seo J. K., and Yun B. K. (2002) `Supply chain simulation with discrete-continuous combined modeling', *Computers & Industrial Engineering*, Vol 43, pp. 375-392.

Martin, Management Accounting Concepts Techniques and Controversial Issues, http://home.att.net/~jvmartin/MAWTextbookMain.htm.

Microsoft Visual Basic Language Reference, 1995, Microsoft Corporation.

Pundoor, G. (2002) `Supply chain simulation models for evaluating the impact of rescheduling frequencies', MS 2002-9, Institute for Systems Research, University of Maryland, College Park. Available online at http://techreports.isr.umd.edu/TechReports/ISR/2002/MS_2002-9/MS_2002-9.phtml.

Simulation Dynamics (2001) `Supply chain builder: a simulation platform for developing understanding of complex supply chains', http://www.simulationdynamics.com/Sc/index.htm.

Supply-Chain Council (2000) *Supply-Chain Operations Reference-model*, Version 4.0, Pittsburgh, Pennsylvania.

Supply-Chain Council (2004) *Supply-Chain Operations Reference-model*, Version 6.1, Pittsburgh, Pennsylvania.

Swaminathan, J.M., Smith, S.F., and Sadeh, N. M. (1998) `Modeling Supply Chain Dynamics: A Multiagent Approach', *Decision Sciences*, Vol 29 (3), pp. 607-631.

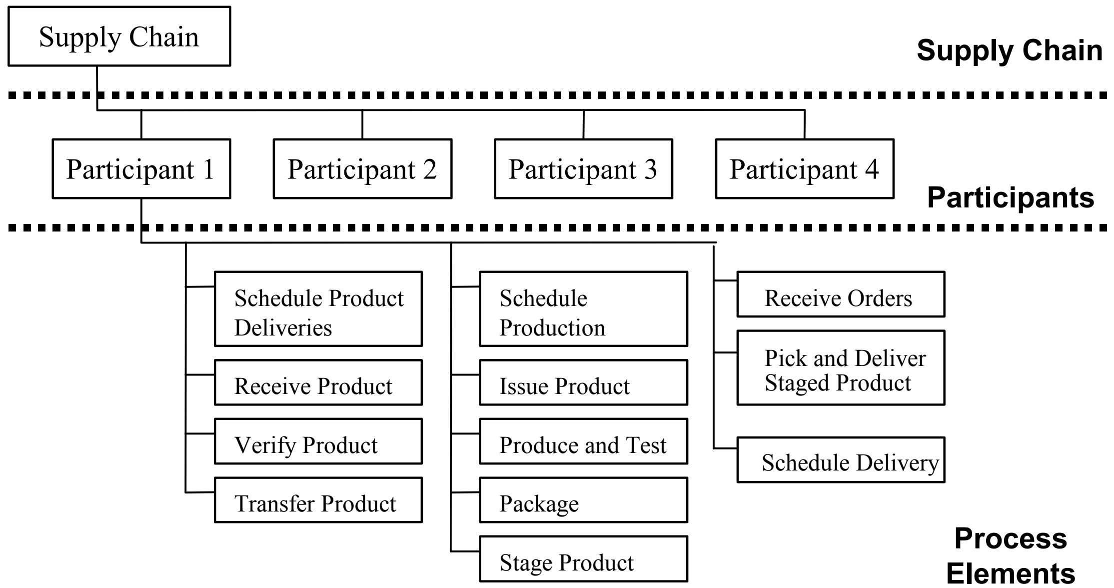


**Figure 1**. *Submodel Hierarchy*

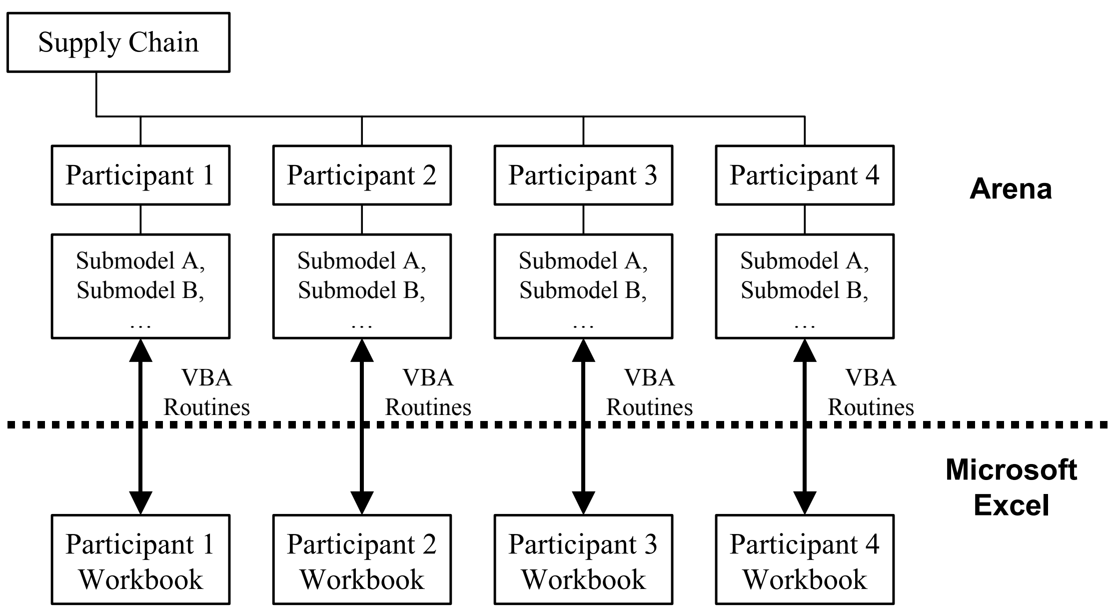


**Figure 2**. *Arena and Excel Integration*

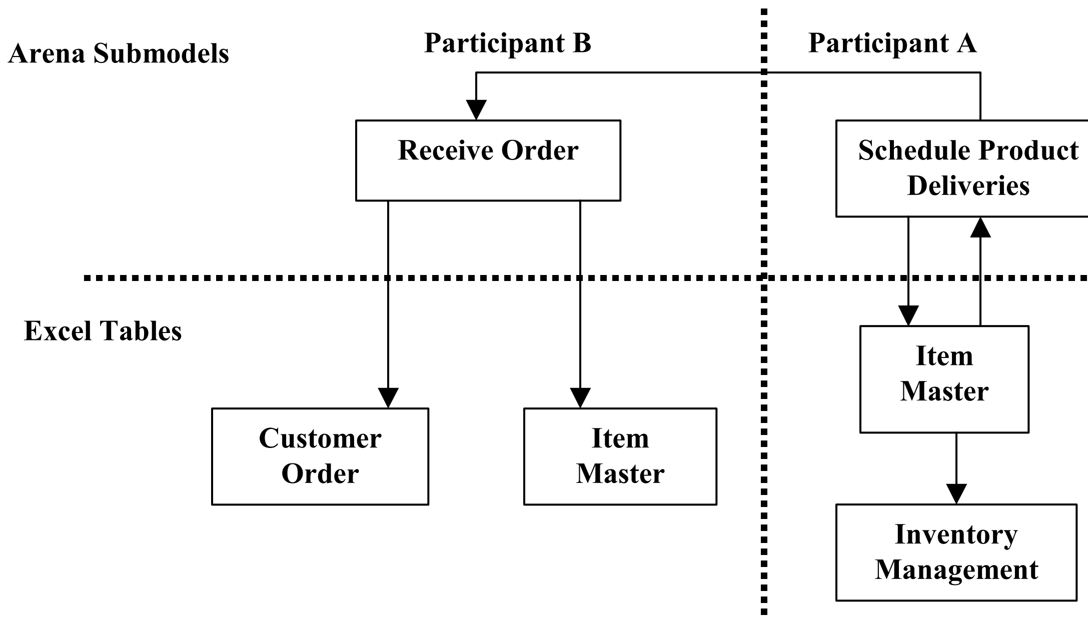


**Figure 3**. *Participant A placing an order with Participant B*

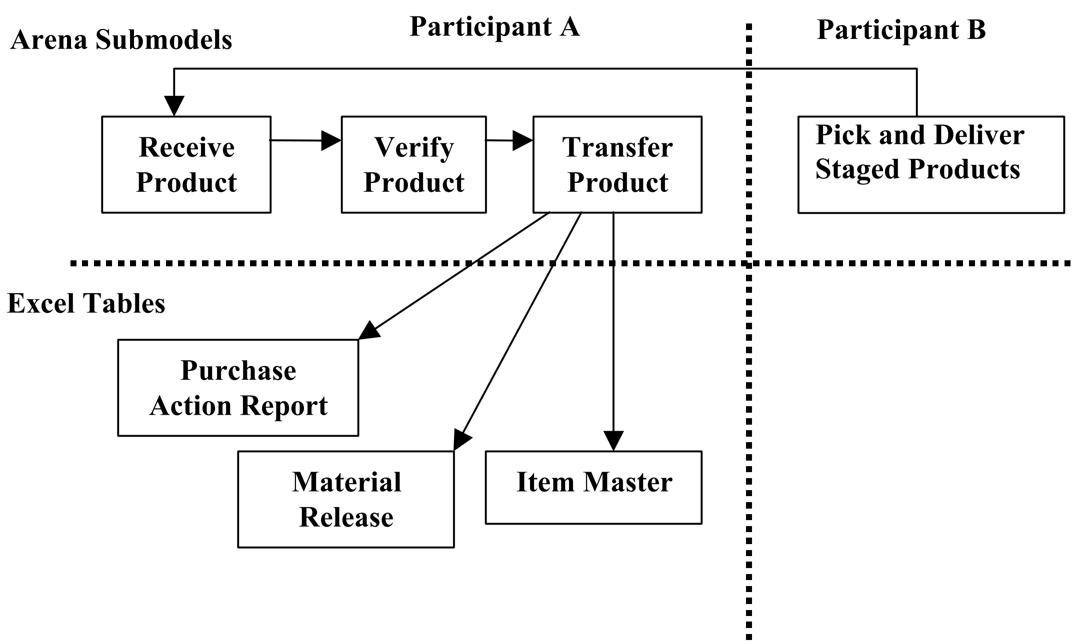


**Figure 4.** *Participant A receiving delivery of orders from Participant B*

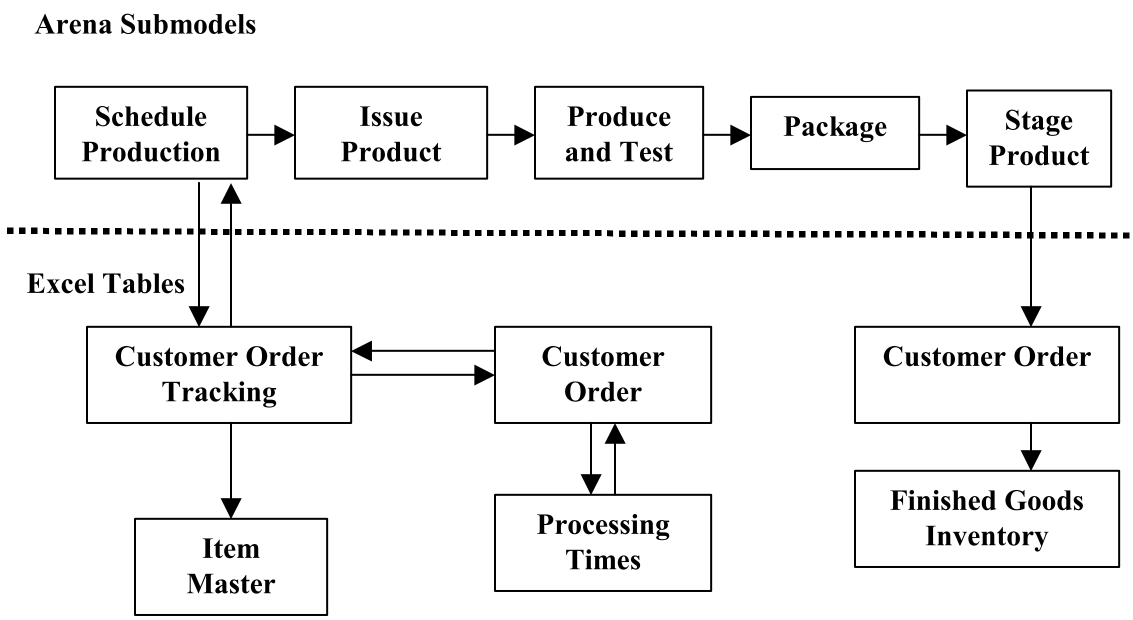


**Figure 5.** *Checking Open Orders for Production*

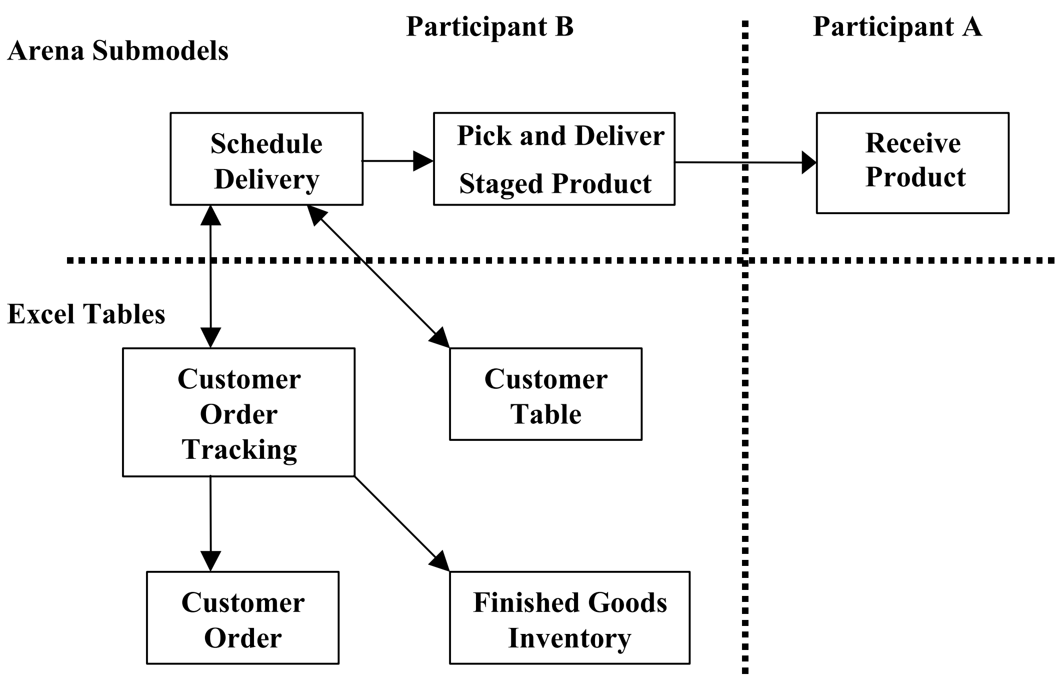


**Figure 6.** *Participant B delivering completed orders to Participant A*

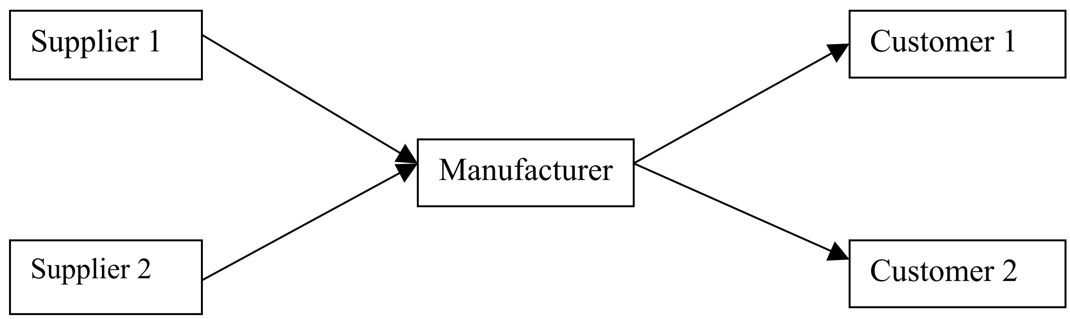


**Figure 7.** *Supply chain network for the first supply chain*

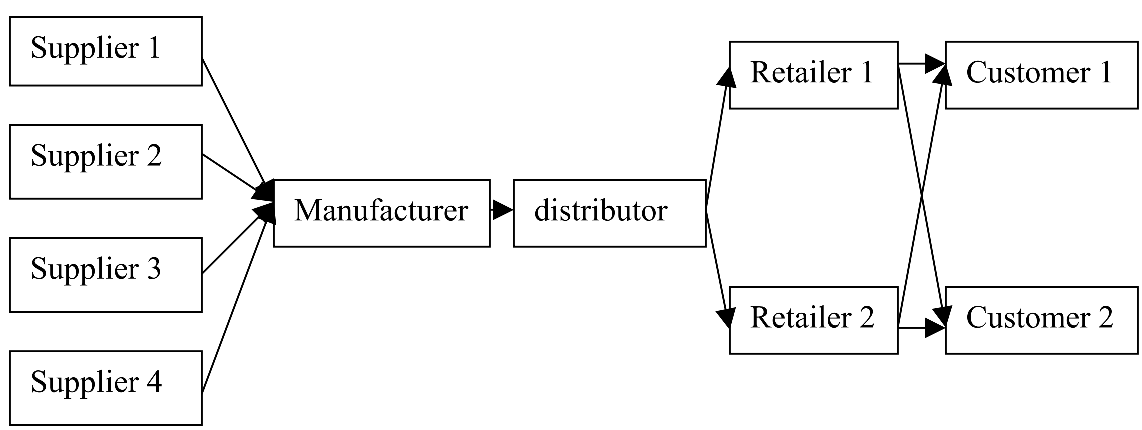


**Figure 8.** *Supply chain network for the second supply chain*