Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

SIMULATION RESULTS FOR SUPPLY CHAIN CONFIGURATIONS BASED ON INFORMATION SHARING

Rong Liu and Akhil Kumar

Dept. of Supply Chain and Information Systems
Penn State University
University Park, PA 16802, U.S.A.

Alan J. Stenger

Ports of Auckland Visiting Professor of Logistics
and Supply Chain, University of Auckland
Auckland, New Zealand

ABSTRACT

As supply chains evolve beyond the confines of individual organizations, information sharing has become the Holy Grail in supply chain technology. Although the value of information sharing is well recognized, there is little research on how to use it to configure supply chains. This paper proposes a parameterized model to capture information sharing in a supply chain. By changing the parameters of this model, we actually adjust information sharing and create supply chain configurations. Configurations are the means of responding to events or changes in supply chains in a timely manner. A complete example is used to demonstrate this methodology. We also perform simulation experiments to compare configurations and to understand the effect of information sharing on supply chain performance. Thus, we show how to achieve supply chain configurability by leveraging information sharing.

1 INTRODUCTION

In recent years, the competitive business environment has forced companies to reduce costs while still providing high quality products and services in great variety and customizability. This challenge has compelled companies not only to optimize the internal logistic functions, but also to build real-time *collaboration* across organizations for mutual gains through information sharing (Finley and Srikanth 2005, NØkkentved and Hedaa 2000).

Research has shown that through information sharing, companies can establish strategic partnerships, coordinate processes, and create efficiencies and cost savings in the entire supply chain (NØkkentved and Hedaa 2000). Moreover, Gosain et al. (2004) showed that information sharing can increase supply chain flexibility, the extent to which supply chain linkages are able to adapt to changing business conditions. In addition, information sharing can lead to new knowledge creation in supply chains (Malhotra et al. 2005).

However, as the level of collaboration increases, shared information tends to be richer and more diverse. A critical issue is how to manage information sharing so that companies have enough visibility about the status of the supply chain, and yet the volume of shared information is not overwhelming (Malhotra et al. 2005). More importantly, shared information is “relevant enough and generated frequently enough so that partners can make decisions that compensate for the inevitable unplanned occurrences” (Finley and Srikanth 2005). This requires supply chains to adjust information sharing (e.g., by relevancy, frequency, aggregation level etc.) in a timely manner in response to various events or exceptions. Such an adjustment may result in changes in supply chain processes, such as changes of activities, changes in activity execution sequences, and new exception handling processes. Malhotra et al. (2005) also pointed out that supply chains need to architect inter-organizational processes to coordinate information exchange. We view such changes as *supply chain configurations* (or simply *configurations*).

Thus, *a supply chain configuration refers to a set of supply chain activities, the specific pattern of inter-organizational linkages and information sharing among them*. In general, supply chain configurations reflect a supply chain’s experience of reacting to events or changes and inferences can be derived from them in response to similar events or changes in the future. In that sense, configurations can be referred to as a part of "organizational memory" (Gosain et al. 2004, Malhotra et al. 2005).

In this paper, we approach the goal of designing supply chain configurations by leveraging information sharing. We use a parameterized model to describe information sharing involved in an inter-organizational process. Then we modify parameters of this model to adjust information sharing and achieve new supply chain configurations. The performance of configurations is evaluated by simulation. When events or changes are sensed, we apply appropriate configurations in response to them.

The remainder of this paper is organized as follows. The next section briefly discusses the related work. Section 3 introduces the parameterized information sharing model. In Section 4, our methodology for configuring supply chains is illustrated by a complete example. Section 5 concludes the paper and briefly describes our planned future work.

2 RELATED WORK

The concept of *supply chain configurations* is introduced in Supply Chain Operations Reference (SCOR) model (Supply-Chain Council 2003). SCOR is a business process reference model that provides a framework for configuring supply chains at the process category level. For example, in a supply chain, the supplier can choose *make-to-stock* process category while the manufacturer uses *make-to-order* one. Such configurations are long-term and have strategic implications. However, they may not be applicable to short-term changes, which typically have an impact on tactical or operational decisions and need real-time responses. Configurations for short-term changes can provide supply chains great *agility* (Lee 2004).

Moreover, research shows that organizational memory allows organizations to recognize types of adjustments needed in response to events or changes (Malhotra et al. 2005). Gosain et al. (2004) proposed a conceptual sense-and-adapt framework for dynamic adjustment with organizational memory. Still, we lack a detailed methodology for developing and utilizing organizational memory for supply chains. In this paper, we create such a methodology based on information sharing. Some other related work includes adaptive enterprises (Haeckel 1999) and a technical framework for sense-and-respond business management (Kapoor et al. 2005).

3 INFORMATION SHARING MODEL

In this section, we describe a modeling approach for information sharing. Supply chain partners need to share various information, including operational information such as inventory status, strategic information such as market trends and production capabilities, and exceptions in order to respond to changes in supply chains promptly and appropriately (Gosain et al. 2004, Malhotra et al. 2005). However, the quality of shared information can be a major concern. There are different dimensions of information quality, including *relevance*, *accuracy*, *completeness*, *timeliness* and *compatibility* (Miller 1996). We propose a parameterized model to capture these dimensions.

We extend Event-Condition-Action (ECA) rules (McCarthy and Dayal 1989) to information sharing. *An ECA rule specifies that when an event occurs and if certain conditions hold, a specific action is executed.* In our context, actions mean sending information flows. Moreover, an information flow can be decomposed into a set of parameters. Therefore, information sharing can be described in terms of the following parameters: *events*, *conditions*, *information flows* (*senders*, *receiver*(*s*), *shared data objects*, *data templates*, *requested recipient actions*, *frequency*, *batch/real-time*, *aggregation levels*). The main advantage of this parameterized approach is that information sharing can be leveraged by adjusting the parameters. The details of this model can be found in (Liu and Kumar 2003). Next, we briefly describe different parameters.

Events are signals for information flows to occur. For example, *changes in shared data objects* are events since they can cause exchange of information flows. In addition, *temporal events* and *exceptions* can also trigger information flows. For example, if weekly demand sharing leads to high forecast errors (exceptions), real-time sharing may be used to improve the forecast precision.

Conditions are a collection of queries on shared data objects. If all shared data objects are XML documents, the queries can be defined using XQUERY (2005).

When an event occurs and specific conditions are satisfied, an associated **information flow** is sent out. This flow can generate an event indicating some changes to shared data objects or prompt the recipient to take action on it, and perhaps a subsequent flow is generated if the corresponding conditions are satisfied. Thus, information flows are linked together by means of events and conditions. Sample flows will be provided shortly.

An information flow has mandatory and optional parameters. Mandatory parameters include *sender* and *receiver*(*s*), *shared data objects* and *templates*. Sender and receiver(s) are the communicating partners. In general, shared data objects should be *relevant* to collaborative scenarios. In a dynamic supply chain, information relevant to one situation may be irrelevant to another. Therefore, information sharing should be analyzed and adjusted in a timely manner. Moreover, shared data should be *accurate* and *complete*. Templates give the formats of data objects, such as EDIFACT and XML.

The following are optional parameters. *Requested recipient action* specifies the actions taken by the recipient after the flow is received. *Frequency* (*Batch/Real-time*) of sending information flows captures the *timeliness* requirement of shared information. *Aggregation levels* can be transactional (e.g. POS data), per item or per brand etc. This parameter further specifies the *relevance* of shared information. More parameters pertaining to describe information flows can also be added when necessary.

4 CONFIGURING SUPPLY CHAINS

4.1 Methodology

In this section, we will discuss how to apply the parameterized model to configure supply chains. First, we need to capture information sharing between partners precisely. Since an inter-organizational process directly involves information sharing, we describe such a process formally using a UML activity diagram (OMG 2003).

We model supply chain activities as actions, and data inputs or outputs of supply chain activities as objects. In addition, we use UML *swimlanes* to distinguish different partners. A detailed example will be provided later. With such an activity diagram, we can immediately recognize information flows and shared data objects involved in this process. Specifically, any object flow from one partner to another can be considered as an information flow, and the object of this flow can be treated as a shared data object. Therefore, using activity diagrams, we can precisely describe the information sharing between partners and then represent it using the parameterized model.

Next, we propose a general methodology that involves the following steps:

1. Describe/modify a process as a UML activity diagram and check if this diagram is correct;
2. Extract cross-swimlane object flows from the UML diagram and save them as parameterized information flows in a table. Adjust parameters to create different supply chain configurations;
3. Check if the new configurations are correct (in terms of parameter values, conditions, etc.);
4. Evaluate configuration performance by simulation;
5. Store the configurations in a standard form such as XML and exchange them with partners.

4.2 Example: Vendor Managed Inventory

Next, we illustrate this methodology using a detailed example: *Vendor Managed Inventory* (VMI). VMI is a collaborative arrangement typically between a vendor and its customers, such as retailers. In VMI, the vendor takes over the replenishment planning task for its partners. The main steps in VMI are: (1) customers share their actual demand or usage with the vendor; (2) the vendor generates demand forecast and places replenishment orders for customers; (3) customers review replenishment orders and confirm them; (4) the vendor then sends ship notices, followed by physical goods transfer; (5) customers acknowledge the actual receipt or return goods; and (6) there may be a need for exception handling when expected performance, such as a 95% order fill rate, is not achieved.

Figure 1 shows the UML activity diagram for this VMI process. This diagram clearly identifies information flows and shared data objects. Object flows which cross swimlanes are information flows and they carry shared data objects. In Figure 1, the seven information flows are denoted by numbers in the sequence in which they occur.

4.3 Supply Chain Configurations

Next, we extract the information flows from the UML activity diagram and store them in a table. This step can be facilitated with automated tools. For example, first, the pictorial UML diagram can be converted into an XML description using conversion tools such as Rational XDE (IBM 2003). Once the process description is available in XML, it can be parsed to extract each individual flow by writing an XML Stylesheet Transformations (XSLT) script and storing it in a *configuration table*. Additional information, not captured by a UML diagram, such as template numbers for shared data and transfer mode, can be added to the table. In addition, one could add the expected delay for each flow, so the actual throughput time could be compared against the expected value. Thus, *the configuration table gives the rules of interaction between partners*. Table 1 shows a configuration table for information flows extracted from the UML diagram of VMI. This table can capture the information sharing involved in the VMI process, and is called *Configuration 1* (***C1***).

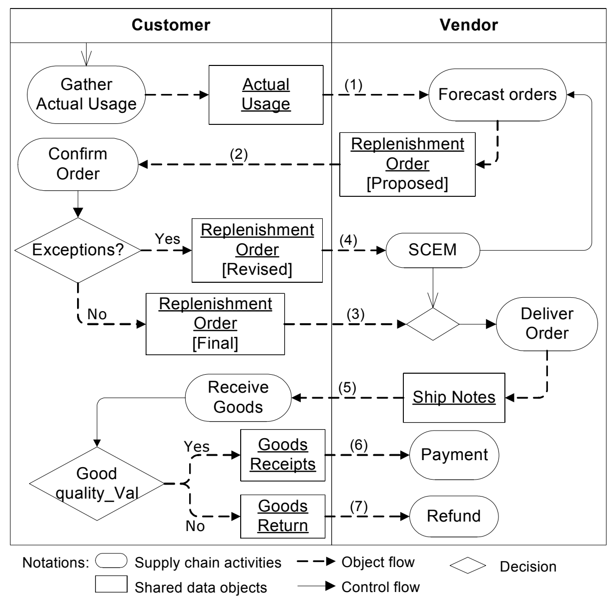

Figure 1: Modeling VMI with UML Activity Diagram

In a configuration table, every information flow is initiated by an event, and takes place upon checking an (optional) associated condition; if the condition is true, then the flow takes place. For example, in Table 1, the first row corresponds to the *sendUsage* information flow. This flow occurs at 5 PM every Monday (a temporal event). Thus, the usage information is sent in a weekly usage form (i.e., a template) from the customer to the vendor. The second row describes the action taken by the vendor on receiving the usage. If the inventory value falls below the reorder point, then a new information flow called *proposeOrder* is sent from the vendor to the customer. The customer either accepts the proposed order (row 3), or rejects it and sends a modified order to the vendor (row 4). Then, after a ship notice is issued (row 5), receiving or returning of goods (rows 6/7) follows.

Table 1: Sample Data in Configuration Table for VMI (Configuration 1)

Information Flow	**Event/ time**	**Condition**	**Action (Send Information Flow)**				**Requested Recipient Action**	**Batch/ Real-time**	**Lead time**
			Sender	**Receiver**	**Data Objects**	**Template**			
(1) sendUsage	Monday, 5 PM	–	Customer	Vendor	Weekly Usage	#852	Propose Order	Batch	0.5
(2) propose-Order	Usage Received	Inventory <*ROP* (Reorder Point)	Vendor	Customer	Repl. Order [Proposed]	#855	Confirm Order (Accept or Reject)	Batch	0.5
(3) accept-Order	Proposed order received	–	Customer	Vendor	Repl. Order [No change]	#855	Generate Ship Notice	Real-time	0.5
(4)modify-Order	Proposed order received & exception (fill rate<*ft*)	–	Customer	Vendor/ SCEM	Repl. Order [Revised]	#855	Generate Ship Notice	Real-time	0.5
(5) ShipNotice	Confirmed Order received	If Shipday = Sat; ship_gnd else ship_air	Vendor	Shipper/ Customer	Ship Notice	#857	Receive goods	Real-time	0.5
(6) Goods-Receipt	Goods received	Quality_ val >= *q*	Customer	Vendor	Goods Receipts ACK	#861	NONE	Real-time	3.0
(7) Goods-Reject	Goods received	Quality_ val < *q*	Customer	Vendor	Goods Return	#862	Refund	Real-time	3.0

In this framework, there are several avenues for configuration. First, changes may be made to the frequencies of flows. For example, say the "event/time" of the first row of Table 1 is changed from "*Friday, 5 PM*" to "*Daily, 5 PM*". This change leads to a new configuration, called *Configuration 2* (***C2***) described in Table 2. With this configuration, the vendor can track the customer's inventory on a daily basis and replenish inventory responsively.

Table 2: Four Configurations

Configuration	**Description**
C1	Weekly information sharing. See Table 1.
C2	Daily information sharing. See Table 1. Change the event/time of row 1 to "everyday, 5 PM".
C3	Mixed information sharing: IF *exception occurs, i.e. fill rate<ft,* **C2** ELSE **C1**.
C4	Daily usage and machine breakdown information is shared; alternative sourcing is used during breakdowns

However, Configuration C2 may increase costs because of more frequent order replenishment. We conjecture that if the fill rate already reaches a satisfactory level, say 95%, real-time information sharing may not be necessary; real-time information sharing is required only when the fill rate is below the 95% level (we say an exception occurs when the fill rate drops below 95%). Therefore, we create *Configuration 3* (***C3***) that mixes weekly and daily information sharing, as shown in Table 2.

Still, many other adjustments may be made to the parameter values. In Table 1, the reorder point (row 2), the target level for the fill rate (row 4), or the quality threshold (row 6) may be changed to a different value. The condition in row 5 allows the customer to configure the shipment mode depends on the ship day. Finally, the formats of documents can also be easily changed by specifying a new template name, if, say one partner modifies its documents. All of the above changes can be made "on-the-fly," while other flows remain unchanged.

Another aspect of configurability relates to the process itself. This may involve modifying an existing flow (i.e. making a change in a parameter value), adding a new flow, or deleting an existing flow. For example, suppose the *Order Delivery* activity is outsourced to a third-party shipper. The vendor shares the quantities and shipping profiles of replenishment orders with the shipper, and the shipper arranges shipment automatically. This change will require a revised UML diagram and this UML diagram will eventually lead to a modified configuration table.

In addition, sharing information about the occurrences of important events, especially exceptions, makes a supply chain agile and able to recover quickly from sudden setbacks (Lee 2004). For example, suppose the vendor may experience serious machine breakdowns, and, as a result, replenishment orders are delayed. If the vendor can notify the customer of the occurrences of such events, the customer can turn to alternative sourcing. This scenario leads to *Configuration 4* (***C4***), as shown in Table 2.

4.4 Simulation

We saw above that the information sharing model can lead to different configurations of a supply chain. Next, we use Arena Simulation Software (Kelton et al. 2003) to evaluate the performance of configurations *C1-C4* (see Table 2). We first simulate and compare *C1*, *C2* and *C3*. Later, we test *C4*, and then compare it with *C2*.

4.4.1 Simulation Setting

The setting of the simulated supply chain process is shown in Table 3. We assume that there is only one product involved in this VMI arrangement. The daily usage at the customer site follows a Gamma distribution with α=1.25 and β=400 (i.e, Gamma(1.25, 400), mean = $\alpha\beta$ = 500, variance $\alpha\beta^2$ = 200000). Tyworth et al. (1996) showed that if the lead time for an item and the demand per unit of time are both stochastic, Gamma distribution is a good choice for the resulting demand during the lead time. Also, Gamma distribution has non-negative values. Moreover, since demand variability (Waller et al. 1999) may have impact on information sharing, we will also test the performance of configurations when demand variability changes. Demand variability is measured by the coefficient of variation (CV), the standard deviation of daily demand divided by the mean. Figure 2 shows the probability density function of the daily demand which follows different Gamma distributions. These distributions have the same mean, i.e. 500, but different standard deviation and therefore different demand variability. Clearly, Figure 2 shows that when α is large, Gamma distribution closely approximates a normal distribution.

Table 3: Simulation Settings

Simulation setting	Values
Daily usage	Gamma(400, 1.25)
Reorder point	3500
Replenishment order size	6000
Lead time of replenishment	5 days

The lead time for a replenishment order is 5 days (see Table 1 for the specific lead time of each information flow). A (*ROP*, *Q*) inventory policy is used, i.e., whenever the vendor knows that the inventory at the customer site is below the reorder point (*ROP*=3500), a replenishment order with order size *Q*=6000 is proposed.

To evaluate a configuration, appropriate performance metrics are chosen (Chopra and Meindl 2001). These are *average flow time* (or *inventory turns*), order *fill rate* and *annual total cost*. *Average flow time* is the time in days it takes to consume the average inventory (i.e., average inventory / average daily sales) and accordingly, *inventory turns* = the number of days in a year / average flow time. We assume 250 business days in a year. *Order fill rate* is defined as the percentage of demand fulfilled by the customer from available inventory.

To calculate the *total cost* of the supply chain, a simple but realistic cost structure is chosen based on a sale price of each item at $1.00 per unit at the customer side (the other costs are proportional to this sales price). Partial fulfillment is allowed, whereas back orders are counted as lost orders. Average shortage cost per lost item is 20% of the sales price, which reflects the cost of lost potential sales opportunities. In addition, average carrying cost per item per year is 20% of the sales price, which reflects the cost of storing and handling the product. Average transportation cost per item is $0.10. Average manufacturing cost per item from the VMI vendor is $0.20. Setup cost for every replenishment order is $100 incurred by order handling and setting up a production run. When a replenishment order is proposed, if the accumulated fill rate is below 90%, a penalty of $1000 is applied because the performance fails to reach the required level (see "modifyOrder" row in Table 1). This penalty reflects the sales loss as a result of customers switching to competitive brands since their needs cannot be satisfied. Thus, *total cost per year = setup cost of replenishment orders+ manufacturing cost + transportation cost + carrying cost + shortage cost + penalty*.

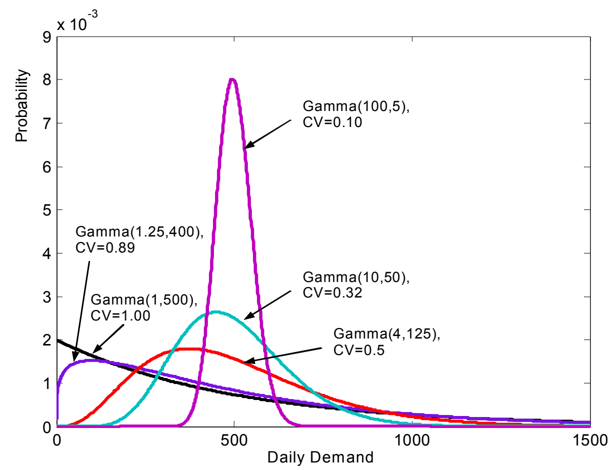

Figure 2: Probability Density of Gamma Distributions

Angulo et al. (2004) used a similar cost structure to test the impact of information accuracy and information delay on supply chain performance in a VMI arrangement. Of course, supply chain scenarios may have different cost structures. To further demonstrate the impact of cost structures on configuration selection, we will provide sensitivity analysis for key cost components later.

4.4.2 Simulation 1 - Comparing Weekly Sharing (C1), Daily Sharing (C2), and Mixed Sharing (C3)

We simulated three configurations *C1-C3* (see Table 2) for 15 replications each for a period of 1000 days. Table 4 shows the performance results of each configuration.

First, the fill rate of *C1* is the lowest among the three configurations. Compared with *C1*, *C2* has a much higher fill rate, almost 100%. This is due to real-time information sharing. However, in *C2*, the average flow time is also increased by 1.5 days. In other words, more inventory is kept in the customer's warehouse because more replenishment orders are placed.

Although there is naturally a trade-off between fill rate and inventory turns, it would be interesting to explore whether it can be fine tuned to achieve a satisfactory fill rate while keeping the inventory turns as high as possible. We believe information sharing is the answer here, and test this belief in configuration *C3*. Recall that in *C3*, weekly sharing and daily sharing are mixed. Daily sharing is used when the fill rate drops below 95%. As Table 4 shows, *C3* realizes not only a satisfactory fill rate, 95%, but also less average flow time, about 0.73 day less than that in *C2*. In *C3*, information is not always shared in real time, but is shared whenever necessary or in "quasi-real time" (Finley and Srikanth 2005).

Table 4: Performance Comparison of *C1*, *C2* and *C3*

Configuration	**Fill rate (%)** (μ ± σ)	**Avg. flow time (days)** (μ ± σ)
C1 (Weekly sharing)	92.28 ± 1.39	**6.54** ± 0.26
C2 (Daily sharing)	**98.49** ± 0.61	8.07 ± 0.16
C3 (Mixed sharing)	95.01 ± 0.26	7.34 ± 0.18

Table 5 compares the total cost per year incurred by each configuration. As Table 5 shows, the total cost of *C2* is lower than that of *C1* because *C2* has much higher fill rate than *C1*, and, as a result, *C2* incurs significantly lower shortage cost and fewer penalties than *C1*. This saving can balance the extra setup, manufacturing, shipping and carrying costs resulting from more inventory required by *C2*. However, although the shortage cost of *C3* is higher than that of *C2*, *C3* still incurs slightly lower total cost than *C2*. Because *C3* keeps less inventory than *C2*, the cost reduction in setup, manufacturing, shipping and carrying inventory can compensate for the extra shortage cost and penalties resulting from lower fill rate in *C3* (3.48% lower than that in *C2*).

Table 5: Cost Comparison of *C1*, *C2* and *C3*

Total Cost Per Year ($)	**Configurations**		
	C1 (Weekly Sharing)	C2 (Daily Sharing)	C3 (Mixed Sharing)
Setup	**1,898**	2,060	1,981
Manufacturing	**23,084**	24,722	23,816
Shipping	**11,542**	12,361	11,908
Carrying	**654**	807	736
Shortage	1,947	**382**	1,254
Penalty	2,533	**17**	300
Total (μ ± σ)	41,659 ± 4,236	40,349 ± 1,056	**39,995** ± 1,039

Configuration *C3* shows that the desired supply chain performance (order fill rate, cost etc.) can also be achieved through flexible information sharing. Moreover, the simulation further suggests that information sharing can be a tool for dynamically adjusting supply chain processes in response to exceptions in supply chains.

4.4.3 Simulation 2 – Sharing Information about Event Occurrences

Next, we simulate the impact of sharing information about the occurrences of machine breakdown events on supply chain performance. Fox et al. (2000) showed that sharing information about unexpected disruptions can enhance the coordination of supply chain partners and reduce the negative consequences of those disruptions.

It is reasonable to expect that machine break downs will occur. During the breakdown, all replenishment orders are delayed until the problem is fixed. The up time of these machines follows an exponential distribution with a mean of 90 days, i.e. EXP(90), and the down time follows EXP(5). With Configuration *C2*, the vendor does not notify the customer of the occurrences of breakdown events, so replenishment orders could be delayed. With Configuration *C4*, the customer is notified when breakdown events occur, and then it turns to alternative vendors for replenishment. The manufacturing cost of alternative vendors is 50% higher than that of the VMI vendor. The lead time of alternative sourcing follows a uniform distribution between 3 and 5 days, i.e., U(3,5). After the machines are fixed, the customer resumes the replenishment activities with the VMI vendor as before. Still, daily usage is shared in both *C2* and *C4* (See Table 2).

Table 6: Performance Comparison of *C2* and *C7*

Performance Indexes	**Configurations**	
	C2 (not sharing breakdown info.) (μ ± σ)	C4 (sharing breakdown info.) (μ ± σ)
Repl. orders from VMI Vendor per year	19.35 ± 0.48	18.25 ± 0.78
Repl. orders from alt. vendor per year	–	1.78 ± 0.74
Fill rate (%)	93.84 ± 2.05	**95.92** ± 0.81
Avg. flow time (days)	**6.98** ± 0.25	7.15 ± 0.17
Total cost per Year($)	41,341 ± 3,278	**41,135** ± 1,411

Next, we can show that with sharing information of breakdown events, the performance of the supply chain improves. As Table 6 shows, in terms of the fill rate, *C4* clearly outperforms *C2*. Moreover, compared with *C2*, *C4* has only slightly increased average flow time. Also, the total cost per year decreases when the customer is notified of the breakdowns, and alternative sourcing is introduced.

Although an extra cost is incurred by alternative sourcing, *C4* leads to lower shortage cost and fewer penalties than *C2* as order fill rate improves.

4.4.4 Sensitivity Analysis

From Table 5, we can see some cost components vary significantly among *C1*, *C2* and *C3*. Next we do sensitivity analysis for *carrying cost*, *shortage cost* and the *penalty*. Finally, we will analyze the sensitivity of the configurations to *demand variability*.

Figure 3 shows the sensitivity analysis of total cost to carrying and shortage costs. If we change the carrying cost per item per year from \$0.10 to \$0.40, but keep the shortage cost per item to \$0.20, we can see *C3* always outperforms the other two. This result can be explained by Table 5, which clearly shows that the carrying cost only amounts to about 2% of the total cost. The change in carrying cost makes no significant impact on the total cost.

On the other hand, if the shortage cost per item varies from \$0.00 to \$0.40, the configuration with the lowest total cost moves from *C3* to *C2*. Clearly, when the shortage cost per item increases, the lower the fill rate, the faster the total cost per year increases.

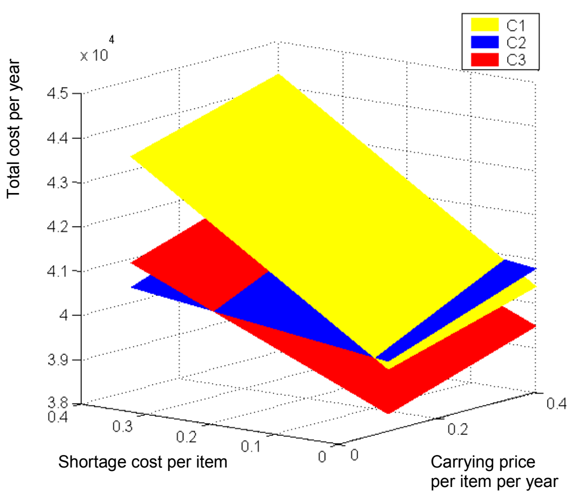

Figure 3: Sensitivity of Total Cost to Shortage and Carrying Costs

Figure 4 shows the sensitivity of the configurations to the penalty imposed when the fill rate is below 90% upon receiving a proposed replenishment order. This figure shows that when the penalty is very small (less than \$300), *C1* incurs the lowest total cost. When the penalty increases, *C3* has the lowest total cost. When the penalty is very high (more than 2300), *C2* incurs the lowest cost since its order fill rate rarely falls below 90%. In general, the penalty represents the cost of losing potential market share because of failures in order fulfillment. In a market with many competitive products, this cost could be very high. Therefore, real-time information sharing is especially important, as this analysis shows.

Next, we test the impact of demand variability on the selection of supply chain configurations. Waller et al. (1999) showed that daily demand variability varies widely in different industries. For example, it is in general lower (around 0.10~0.30) in consumer products and significant higher (perhaps greater than 1.00) in electronics. We tested five daily demand distributions where the demand variability ranges from 0.10 to 1.00, as shown in Figure 2.

Figure 5 shows the total cost change of *C1*, *C2* and *C3* as the demand variability changes. When the demand variability is low, even weekly information sharing (*C1*) can achieve a high fill rate (above 95%), and it requires relatively lower level of inventory than daily information sharing (*C2*). As a result, *C1* incurs lower total cost than *C2*. However, when the demand variability is high, only more frequent information sharing can ensure a high fill rate. Obviously, the total cost of *C2* becomes lower than that of *C1*. An interesting result is that *C3* can always achieve the lowest total cost when the demand variability increases. Recall that *C3* is a mix of weekly and daily sharing and the portion of weekly or daily sharing is adjusted by the fill rate. When the demand variability increases, the portion of weekly sharing is reduced but that of daily sharing is increased. In other words, *C3* approximates to *C1* when the demand variability is low but moves close to *C2* when it is high. Therefore, *C3* can always balance the shortage cost and the replenishment cost and incur the lowest total cost.

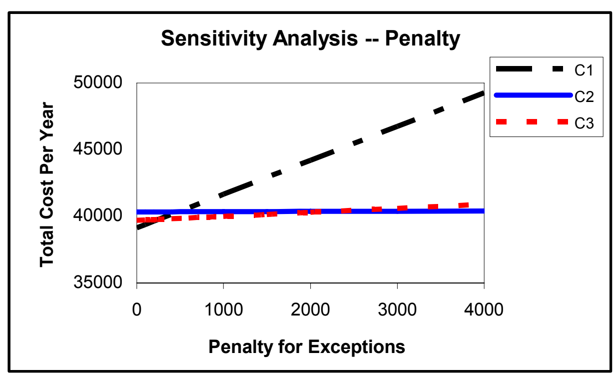

Figure 4: Sensitivity of Total Cost to Penalty

The sensitivity analysis further suggests that in order to achieve the lowest cost in a supply chain, the configurations should be carefully evaluated. Changes in supply chain environment could make a previously optimal configuration no longer optimal. For example, if the shortage cost per item is increased to above \$0.50 (say, because of shortage, ultimate customers lose goodwill and potential sales are lost), clearly, a high fill rate is preferred and real time information sharing becomes necessary, as shown in Figure 3. In addition, the changes in penalty mechanisms and demand variability can also affect the performance of a configuration. In general, changes in supply chains can result in different information sharing needs and suitable configurations should be used accordingly.

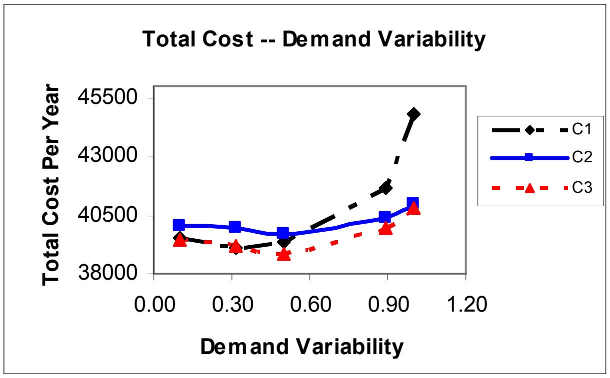

Figure 5: Sensitivity of Total Cost to Demand Variability

5 CONCLUSION AND FUTURE WORK

Information sharing plays a key role in supply chain collaboration, which requires timely information about suppliers, manufacturing, distribution, retailing, and demand. In this paper, we introduced a methodology which leverages information sharing to configure supply chains based on well-known technologies including UML, XML and ECA rules. This methodology consists of several steps, many of which can be automated (or partially automated) using existing tools. Through this methodology, we are able to analyze information sharing, create supply chain configurations, evaluate configurations and use them suitably in response to supply chain changes.

We showed that supply chain changes (e.g., changes in cost structures, market competitiveness and demand variability, etc.) and exceptions can lead to different information sharing requirements and then suitable configurations should be selected to meet the requirements. The results of the simulation show that well-designed configurations can lead to improved performance of a supply chain. In addition, all configurations are shared among supply chain partners and, per se, create new knowledge or "organizational memory" (Malhotra et al. 2005).

We expect our future work to extend this methodology to the strategic level in designing supply chains, and also to focus on the implementation of our methodology.

REFERENCE

Angulo, A., H. Nachtmann, and M. Waller. 2004. Supply Chain Information Sharing in a Vendor Managed Inventory Partnership. *Journal of Business Logistics* 25 (1): 101-116.

Chopra, S., and P. Meindl. 2001. *Supply Chain Management*: *Strategy, Planning, and Operations*, Prentice Hall, Upper Saddle River, NJ.

Finley, F., and S. Srikanth. 2005. 7 Imperatives for Successful Collaboration. *Supply Chain Management Review*, January/February, pp. 30-37.

Fox, M. S., M. Barbuceanu, and R. Teigen. 2000. Agent-Oriented Supply Chain Management. *The International Journal of Flexible Manufacturing Systems* 12 (2-3): 165-188.

Gosain, S., A. Malhotra., and O. A. El Sawy. 2004. Coordinating for Flexibility in e-Business Supply Chains. *Journal of Management Information Systems*, 21 (3): 7-45.

Haeckel, S. H. 1999. *Adaptive Enterprise: Creating and Leading Sense-and-Respond Organizations*, Harvard Business School Press.

IBM Rational XDE. Version 2003.06.12. http://www-306.ibm.com/software/awdtools/developer/rosexde/

Kapoor, S., K. Bhattacharya, S. Buckley, P. Chowdhary, M. Ettl, K. Katircioglu, E. Mauch, and L. Phillips. 2005. A technical framework for sense-and-respond business management. *IBM Systems Journal* 44 (1): 5-24.

Kelton, D., R. Sadowski, and D. Sturrock. 2004. *Simulation with Arena*. Mc Graw Hill, New York.

Lee, H. 2004. The Triple-A Supply Chain *Harvard Business Review*, October, pp.102-112.

Liu, R., and A. Kumar. 2003. Leveraging Information Sharing to Increase Supply Chain Configurability. In *Proceedings of International Conference on Information Systems* (ICIS 2003), Seattle, pp.523-536.

Malhotra, A., S. Gosain, and O. A. El Sawy. 2005. Absorptive Capacity Configurations in Supply Chains: Gearing for Partner-enabled Market Knowledge Creation. *MIS Quarterly* 29 (1): 145-187.

McCarthy, D. R., and U. Dayal. 1989. The Architecture of an Active Database System. In *Proceedings of ACM SIGMOD Conference on Management of Data*, J. Clifford, B. G. Lindsay, and D. Maier (eds.). ACM Press, New York, pp. 215-224.

Miller H. 1996. The Multiple Dimensions of Information Quality," *Information System Management* 13 (2): 79-82.

NØkkentved, C., and L. Hedaa. 2000. Collaborative Processes in esupply Networks. In *Proceedings of the 16th International Marketing and Purchasing Group Conference*, U.K, Available via http://www.bath .ac.uk/imp/pdf/90_NoekkentvedHedaa.pdf.

OMG. 2003. Unified Modeling Language Specification, Version 1.5. Object Management Group, (Available via http://www.omg.org/technology/documents/ formal/ uml.htm)

Supply-Chain Council. 2003. Supply-Chain Operations Reference-Model, Overview of SCOR, Version 6.0., Available at <http://www.supply-chain.org/scoroverview.asp>.

Tyworth, J. E., Y. Guo, and R. Ganeshan. 1996. Inventory Control Under Gamma Demand and Random Lead Time. *Journal of Business Logistics*, 17 (1) :291-304.

Waller, M, M. E. Johnson, and T. Davis. 1999. Vendor Managed Inventory in the Retail Supply Chain. *Journal of Business Logistics*, 20 (1):183-203.

XQuery 2000. XQuery 1.0: An XML Query Language. W3C. Available at <http://www.w3.org/TR/2005/WD-xquery-20050404/>.

AUTHOR BIOGRAPHIES

RONG LIU is a Ph. D. student in Business Administration at The Pennsylvania State University. Her research interests include business process modeling, verification and simulation. Her email address is <rliu@us.ibm.com>.

AKHIL KUMAR is a professor of Information Systems at the Smeal College of Business at The Pennsylvania State University. He has a Ph.D. from Berkeley, and has published more than 60 papers in the areas of database systems and structures, data replication, machine learning and workflow systems in leading academic journals, and international conferences and workshops. He serves on the editorial boards for Information Systems Research, INFORMS Journal on Computing, and Information Technology and Management. His current research interests are in design, analysis and verification of workflow processes, design and implementation of e-services, and web-based technologies. His e-mail address is <akhilkumar@psu.edu>.

ALAN J. STENGER is a professor of Supply Chain Management at the Smeal College of Business at The Pennsylvania State University. Currently he serves as Ports of Auckland Visiting Professor of Logistics and Supply Chain Management in the University of Auckland Business School, New Zealand. He has engaged in a wide range of educational and consulting activities both in the United States and abroad. This includes work with companies in the industries such as the chemical, pharmaceutical, consumer packaged goods, glass, information technology, and railroad. His research interests focus primarily on the organization and management of supply chain and logistical activities in manufacturing and merchandising firms, with particular emphasis on the use of decision support and information technology as enablers. His email address is <ajs@psu.edu>.