Inf Syst Front (2011) 13:139–151
DOI 10.1007/s10796-009-9222-8

Leveraging information sharing to configure supply chains

Rong Liu · Akhil Kumar

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 the degree of information sharing and create new 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. A supply chain architecture which allows agility, adaptability and alignment of partner interests is also proposed based on this methodology.

Keywords Information sharing · Coordination · Supply chain configurability · Configurations · Information flows · Supply chain event management · Simulation

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

---

This research was supported in part by IBM Corporation and a substantial part of it was conducted when the first author was at Penn State University.

R. Liu (✉)
IBM T.J. Watson Research Center,
19 Skyline Drive,
Hawthorne, NY 10532, USA
e-mail: rliu@us.ibm.com

A. Kumar
Department of Supply Chain and Information Systems,
Penn State University,
University Park, PA 16802, USA
e-mail: akhilkumar@psu.edu

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 with a complete example, while Section 5 follows up with results of simulation experiments on various configurations. Section 6 discusses implications of this research to supply chain management practices, its limitations, and our future work. Finally, Section 7 concludes the paper.

2 Related work

The concept of *supply chain configurations* is introduced in the 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 a *make-to-stock* process category while the manufacturer might use *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 that permit short-term changes can provide supply chains greater *agility* (Lee 2004). Thus, our focus is on building a sense-and-respond capability of reacting rapidly to short-term changes.

In one stream of research, empirical studies were conducted to classify configurations of inter-organizational relationships (Bensaou and Venkatraman 1995; Malhotra et al. 2005). Bensaou and Venkatraman (1995) developed a conceptual model of inter-organizational relationships based on the fit between information processing needs and information processing capabilities. Based on this model, they empirically discovered five configurations of inter-organizational relationships. Malhotra et al. (2005) also provided a conceptual framework to uncover five supply chain partnership configurations in terms of partner-enabled market knowledge creation. Although these studies can provide insights on the impact of supply chain partnerships at a conceptual level, there is still a need for an operational framework which enables supply chains to evaluate different partner relationships in terms of supply chain performance. In contrast, our aim is to *develop* a methodology for supply chain information sharing strategies, *evaluate* the strategies in terms of operational efficiency, and then build desirable partner relationships to *deploy* these strategies.

Another significant research stream has been directed towards understanding incentives for information sharing in supply chains. Chen (2003) surveyed several papers that try to qualify the value of information in different supply chain settings. Various mathematical models are proposed to study the value of sharing point-of-sale, inventory, cost, lead time, capacity, and other information that is private to one supply chain partner. These models present different information sharing strategies used in supply chain configurations. A partner in a dynamic supply chain may have to adjust its information sharing strategies as its role in the ecosystem evolves. This gives rise to a need for a methodology that can pool these strategies, evaluate them, apply them into business processes to design supply chain configurations, and also allow dynamic switch from one configuration to another. This paper offers such a methodology. To demonstrate the methodology, we use simulation as a convenient way to compare several information sharing strategies. In addition to simulation, these strategies could also be analyzed using theoretical models (Chen 2003), but such theoretical analysis is out of the scope of this paper.

Also, supply chain configurations can serve as organization memories. Research shows that organizational memory allows organizations to recognize types of adjustments needed in response to events or changes (Malhotra et al. 2005). Haeckel (1999) proposed the concept of "adaptive enterprises" that enterprises need to continue self reengineering to adapt to changes. Kapoor et al. (2005)

developed a technical framework for sense-and-respond business management based on supply chain event monitoring and analysis. Gosain et al. (2004) gave 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.

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.

In the scope of this work, *events* are signals for information flows to occur or for effecting changes in supply chain configurations. There are two types of events, *primitive* and *composite* (Liu et al. 2007). Primitive events are captured directly during business process execution. An example of a primitive event is the receipt of a new order, and it can cause information flows to take place. Moreover, *temporal events* are another kind of primitive events which can trigger information flows. For example, two parties might agree to share demand information every Monday at 9 AM. A composite event is an aggregation of several primitive events. Such aggregation could reveal important business information, such as large forecasting errors resulting from a drop in actual usage. Such events can also trigger information flows and often indicate the need for reconfiguring a supply chain. For example, if weekly demand sharing leads to high forecasting errors, then daily (or even real-time) sharing may be used to improve the precision of the forecasts.

Conditions are a set of constraints that operate on shared data objects. Conditions can be checked by means of database queries. Query languages vary depending upon the structure of the data. Thus, if all shared data objects are XML documents, the queries can be defined using XQUERY (2007). For example, the query “find all items in inventory whose level is less than 3,500” can be represented simply by an XQUERY statement: “inventory/item[quantity <3,500]”. Here item is a sub-element of the inventory element, and quantity is one of its attributes.

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 (EDI 1997) and XML. For instance, XML is a well-known standards and it could be a good option for providing data compatibility.

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 describing 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. Typically, it is not very straightforward to capture information sharing and the related details directly. A better approach is to derive information sharing models from supply chain processes. A supply chain process contains intra-organizational sub-processes that are internal to a particular partner, and inter-organizational sub-processes that span multiple partners. Those inter-organizational processes directly involve information sharing. Therefore, we describe such processes formally using UML activity diagrams (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 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 the performance of each configuration by simulation or other appropriate approaches;
5. Store the configurations in a standard form such as XML and exchange them with partners.

Since it is only based on a business process model this methodology is quite flexible in its applicability, process modeling being a well-accepted practice in supply chains (Supply-Chain Council 2003; Li et al. 2002). In addition, this methodology also allows flexibility in adjusting the parameterized information flow table in Step 2 to make it fit with various supply chain scenarios. Moreover, new parameters can also be added as needed to precisely describe information flows occurring in a supply chain scenario.

Next, we illustrate every step of this methodology using a classical supply chain arrangement of *Vendor Managed Inventory* (VMI). However, it should be noted that this methodology can also be applied to newer supply chain models such as Collaborative Planning, Forecasting, and Replenishment (CPFR) (VICS 2002).

4.2 Example: Vendor managed inventory

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 Fig. 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. Moreover, cost estimation could be associated with each flow to calculate the total cost of information sharing. 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***).

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 from the customer to the vendor as a standard EDI document denoted as #852 (Product Activity Data) (EDI 1997). The second row describes the

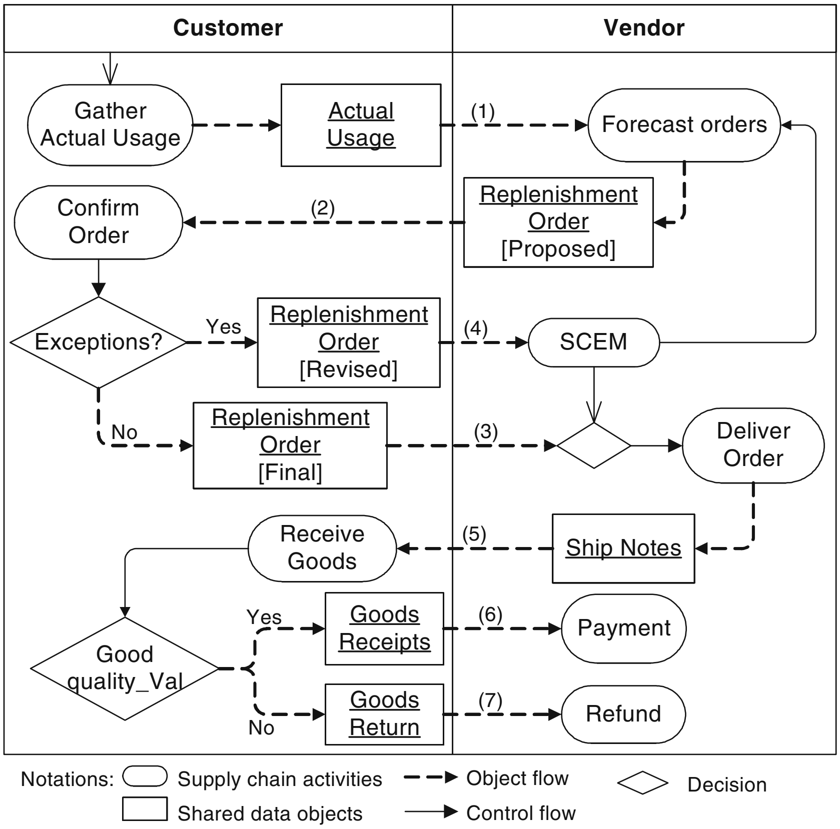

Fig. 1 Modeling VMI with UML activity diagram

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.

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.

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. For example, the reorder point may be adjusted as the demand variability (Waller et al. 1999) or the uncertainty level of demand changes. Demand variability can be measured by the coefficient of variation (*CV*), the standard deviation of daily demand divided by the mean. As a variant of configuration *C2*, we can create a new configuration ***C4*** where the demand is still shared daily, but the reorder point is adjusted dynamically. In *C4*, when demand variability is high, say $CV>1.0$, reorder point is set to a higher value in order to keep more safety stock (reorder point = safety stock + average demand in lead time); otherwise, the reorder point is kept lower. 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.

Table 1 Sample data in configuration table for VMI (configuration 1)

Information flow	Event/tihme	Condition	Action (Send Information Flow)				Requested recipient action	Batch/ real-time	Lead time
			Sender	Receiver	Data Obj.	Template			
(1) sendUsage	Monday, 5 PM	–	Customer	Vendor	Weekly usage	#852 (Product Activity Data)	Propose order	Batch	0.5
(2) propose-Order	Usage received	Inventory<*ROP* (Reorder Point)	Vendor	Customer	Repl. order [proposed]	#855 (Purchase Order Ack.)	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		Vendor	Shipper/ Customer	Ship notice	#857 (Shipment Notice)	Receive goods	Real-time	0.5
(6) Goods- Receipt	Goods received	Quality_ val >=*q*	Customer	Vendor	Goods receipts ACK	#861 (Acceptance Certificate)	NONE	Real-time	3.0
(7) Goods-Reject	Goods received	Quality_ val<*q*	Customer	Vendor	Goods return	#895 (Return Ack.)	Refund	Real-time	3.0

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 experienced 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 ***C5*** shown in Table 2.

In the next section we report the results of experiments that were conducted to evaluate various configurations.

5 Simulation experiments

We saw above that the information sharing model can lead to different configurations of a supply chain. However, each configuration may perform differently under certain supply chain environments. A critical step of our framework is to thoroughly evaluate each configuration and to understand in what circumstances it can perform the best. Both simulation and analytical approaches can be used to analyze configuration performance. To illustrate how to evaluate the configurations extensively, next, we discuss the use of Arena Simulation Software (Kelton et al. 2004) to simulate the behavior of configurations *C1*–*C5* (see Table 2) and provide a detailed analysis. We first simulate and compare *C1*, *C2* and *C3*. Later, we test *C4* and compare it with *C1*–*3*. Finally, *C5* is tested and compared with *C2*.

5.1 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 $\alpha\beta^2$=200,000). 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. Figure 2 shows the probability density function of 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 (measured by *CV*, *CV* = standard deviation / mean of daily demand). Clearly, Fig. 2 shows that when α is large, Gamma distribution closely approximates a normal distribution.

Table 2 Five 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 information sharing with adjustable reorder point: the reorder point is determined by demand variability
C5	Daily usage and machine breakdown information is shared; alternative sourcing is used during breakdowns

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 ($ROP=3{,}500$), a replenishment order with order size $Q=6{,}000$ 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 $1,000 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:

$$\begin{aligned} \textit{Total cost per year} = {} & \textit{setup cost of replenishment orders} \\ & + \textit{manufacturing cost} \\ & + \textit{transportation cost} + \textit{carrying cost} \\ & + \textit{shortage cost} + \textit{penalty} \end{aligned}$$

Table 3 Simulation settings

Simulation setting	Values
Daily usage	Gamma(400, 1.25)
Reorder point	3,500
Replenishment order size	6,000
Lead time of replenishment	5 days

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.

5.2 Results of simulation experiments

In this section, we describe the results of simulation experiments for three different scenarios.

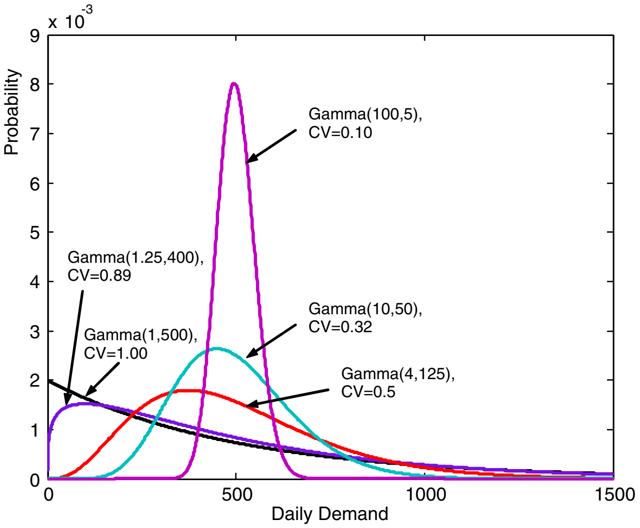

Fig. 2 Probability density function of demand distributions

5.2.1 Scenario 1—comparing weekly sharing (C1), daily sharing (C2), and mixed sharing (C3)

We simulated three configurations *C1–C3* of Table 2 for 15 replications, each for a period of 1,000 days. Table 4 shows the performance results for each configuration.

First, the fill rate of *C1* is the least 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 a reduced average flow time of 0.73 day less than for *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 5 compares the total cost per year incurred by each configuration, and shows that the total cost of *C2* is lower than for *C1* because *C2* has a much higher fill rate than *C1*. As a result, *C2* incurs a 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 a lower fill rate in *C3* (3.48% lower than that in *C2*).

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.

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 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

Also, based on this experiment, one may question whether, as the frequency of information sharing increases and the uncertainty of demand decreases, the safety stock should be reduced accordingly. Therefore, with a careful reduction in the safety stock (i.e. ROP in row 2 of Table 1), *C2* may be superior to other configurations. Next, we perform another experiment to compare configurations with variable information sharing frequency, demand variability, and reorder point.

5.2.2 Scenario 2—comparing C1, C2, C3 and C4 under demand uncertainties

Waller et al. (1999) showed that the daily demand variability differs 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 configurations *C1–C4* under five daily demand distributions where demand variability ranged from 0.10 to 1.00, as shown in Fig. 2. Recall that in configuration *C4* (see Table 2), demand information is shared daily, and, as demand variability increases, the reorder point is also increased accordingly. Further, the reorder point for *C4* is a function of *CV* as per Table 6.

Figure 3 shows the performance of configurations *C1–C4* under different demand distributions. First, compared with *C2*, *C4* incurs less total cost, achieves higher inventory turns, and has lower order fill rate as a result of reducing

Table 6 Configuration *C4*—varying ROP by demand variability

Demand variability (CV)	Safety stock	Reorder point (ROP)
CV> 1.0	2 days	3,500
0.5<CV<=1.0	1 day	3,000
CV<=0.5	0 day	2,500

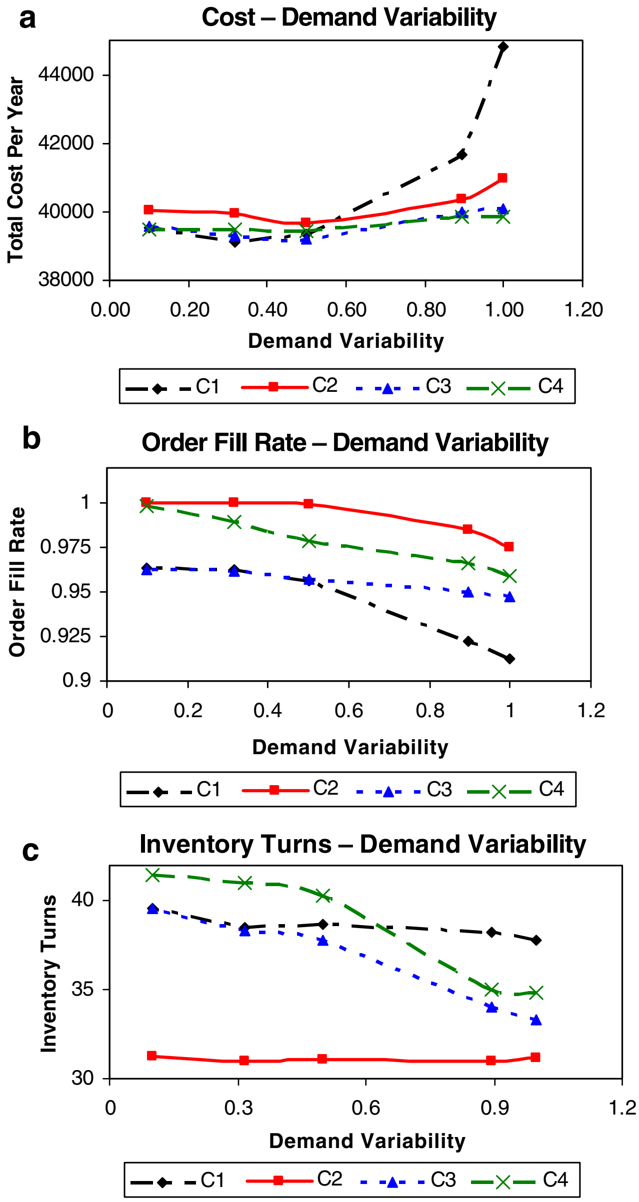

Fig. 3 Comparing configurations C1–C4

safety stock. In particular, when demand variability is not very high (say *CV*<1.0), *C4* outperforms *C2* in the sense that it keeps the total cost low while still maintaining satisfactory order fill rate (higher than 95%) and fast inventory turns. Second, from Fig. 3(a), one can see that when the demand variability is low (say less than 0.5), configurations *C1*, *C3* and *C4* all incur a relatively low total cost. As such, 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 a lower total cost than *C2*. However, as the demand variability grows, more frequent information sharing is required and more safety stock should be maintained to ensure a high fill rate. An interesting observation is that *C3* performs quite close to *C4*, and can always match *C4*, even though 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 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 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.

This experiment shows that when the demand variability is low (*CV*<0.5), the choice of a configuration makes very little difference to performance. In this experiment, weekly information sharing, mixed daily and weekly information sharing, and daily information sharing with an adjustable reorder point can all achieve comparable performance. However, as the demand variability increases, real time information sharing plays a more important role and other parameters, such as reorder point, should be carefully chosen.

5.2.3 Scenario 3—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 breakdowns will occur. Moreover, during a breakdown, all replenishment orders are delayed until the problem is fixed. For our experiments, 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 *C5*, 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 *C5* (See Table 2).

Next, we can show that with sharing information of breakdown events, the performance of the supply chain improves. As Table 7 shows, in terms of the fill rate, *C5* clearly outperforms *C2*. Moreover, compared with *C2*, *C5* has only slightly increased average flow time. Also, the

Table 7 Performance comparison of *C2* and *C5*

Performance indexes	Configurations	
	C2 (not sharing breakdown info.) (μ±σ)	C5 (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

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, *C5* leads to lower shortage cost and fewer penalties than C2 as order fill rate improves.

5.3 Sensitivity analysis

From Table 5, we can see that some cost components vary significantly among *C1*, *C2* and *C3*. Next we do sensitivity analysis for *carrying cost*, *shortage cost* and the *penalty*. Also, Simulation 2 can be treated as an analysis of configuration sensitivity to demand variability.

Figure 4 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, *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.

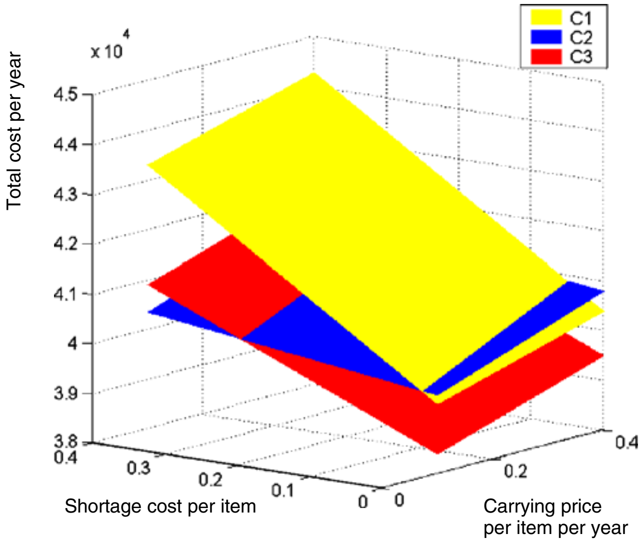

Fig. 4 Sensitivity of total cost to shortage and carrying costs

On the other hand, if the shortage cost per item varies from $0.00 to $0.40, the least cost configuration shifts 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 5 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 2,300), *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.

The sensitivity analysis further suggests that in order to achieve the least 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 was shown in

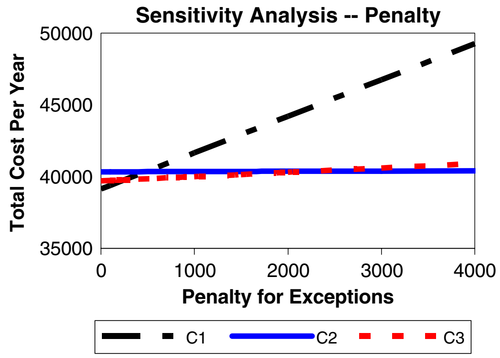

Fig. 5 Sensitivity of total cost to penalty

Fig. 3. In addition, the changes in penalty mechanisms and demand variability can also affect the performance of a configuration.

5.4 Configuration rules

In general, changes in supply chains can result in different information sharing needs and suitable configurations should be used accordingly. For example, Table 8 summarizes some sample supply chain conditions under which our configurations (*C1–C5*) can achieve high performance. This table can be viewed as a set of *business rules for configuring supply chains*. These rules can be used to choose appropriate supply chain configurations, when the supply chain environment causes the conditions to change. For instance, as we analyzed through simulation, *C1* (weekly information sharing) can be chosen when demand variability is very low, shortage cost is negligible, the supply is stable without frequent disruption events, and the expected fill rate is medium. On the other hand, when demand variability becomes unsteady, *C3* or *C4* may be preferable because they are relatively insensitive to demand variability.

6 Managerial implications and limitations

In this section we discuss managerial implications and limitations of this work, and also present an architecture to realize our methodology. The focus of supply chain management has shifted from efficiency and cost-effectiveness to sustainable advantages. Lee's (2004) study shows that supply chains with outstanding performance actually possess three critical qualities: agility, adaptability and alignment (i.e., the ability to align the interests of all supply chain partners). To foster these three capabilities, a supply chain needs to sense changes in supply and demand in a timely manner, interpret these changes, and respond to them quickly by modifying strategies pertaining to supply, products and technologies. In addition, the alignment of partner interests can only be achieved by extensive collaboration and sufficient information sharing. However, traditional ERP systems or supply chain management applications, such as SAP, are typically built upon a set of process reference models (for example, event-driven process chain models are used in SAP (Aalst 1999)), which usually allow very limited configurability. Therefore, still, most supply chain architectures (i.e., supply chain management systems as well as closely related applications and tools) lack agility, adaptability and alignment.

We have tried to develop a supply chain architecture in Fig. 6 that can address these needs. It has several noteworthy features. First, the ERP systems or supply chain applications feed planning, transactional and process-related data into a database which stores shared data objects. Second, an information flow engine based on ECA rules (McCarthy and Dayal 1989) is used to facilitate information flows exchanged between partners. Third, to build "sense-and-respond" capabilities (Kapoor et al. 2005) in a supply chain, an event engine (Liu et al. 2007) is included to detect, analyze, and respond to events in real time. When fed with events from an information flow engine, the event engine can process and filter primitive events, and generate alerts or notifications for only the significant ones. Based on these events the supply chain configuration may be modified suitably.

Finally, as feedback, the changes in supply chain configurations will indicate the adjustment required in the ERP systems or the supply chain applications, in the form of modifications to business processes or system reconfiguration. Kapoor et al. (2005) designed a similar infrastructure to support a sense-and-respond capability. Their architecture enables an enterprise proactively monitor trends in demand and react in a timely manner by optimization techniques. On the other hand, our infrastructure focuses on how information sharing can be leveraged to achieve sense-and-respond capability in a supply chain. Certainly, optimization techniques can also be included in

Table 8 High performance conditions for configurations *C1–C5*

Configurations	Conditions for high performance			
	Fill rate expectation	Demand variability	Shortage cost	Frequency of supply disruption events
C1: Weekly information sharing	Medium (e.g. 90~95%)	Very low (e.g. <0.5)	Very low	Low
C2: Daily information sharing.	Very high (e.g. >95%)	Medium to very high (e.g. >0.5)	High	Low
C3: Mixed information sharing	High (e.g. 95%)		Low to medium	Low
C4: Daily information sharing with adjustable reorder point	Very high (e.g. >95%)		High	Low
C5: Daily usage and machine breakdown information shared	Very high	Medium to very high (e.g. >0.5)	High	High

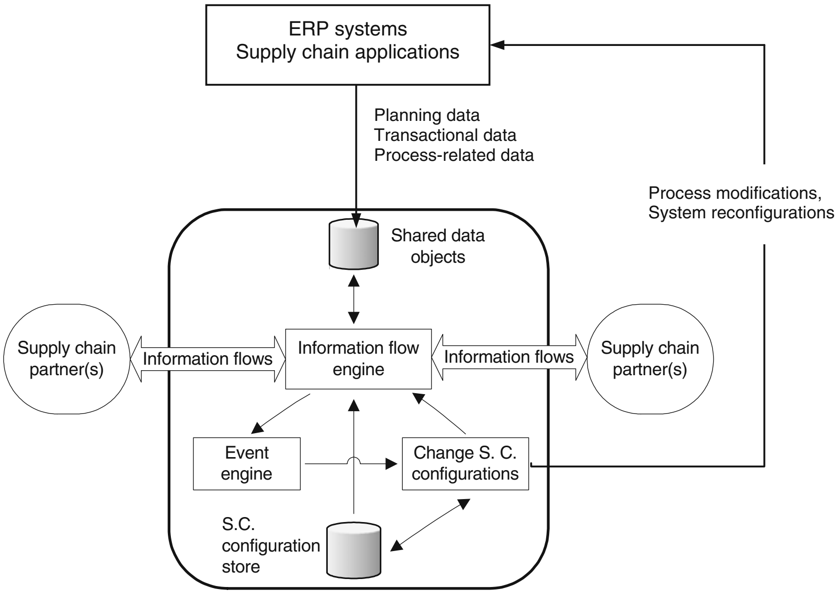

Fig. 6 A supply chain architecture

our architecture as a new module, for example, to discover demand changes, performance exceptions or event patterns by mining shared data objects. Such a module can enhance our architecture to support decision making on a wider range of supply chain problems.

We certainly recognize some limitations of this research and the opportunities it presents for future work. First, we limited our framework for configuring supply chains to the operational and tactical levels. Strategic configurations of supply chains were not explored. Strategic configurations would be relevant when companies make more fundamental decisions, such as those involving a choice between make versus buy, or operating in a brick-and-mortar versus internet environment, etc. For example, the move from a traditional bookstore to online book distribution would definitely require major reconfiguration of the supply chain, and also of the information sharing patterns. Extending this framework to encompass strategic supply chain configurations is left as a future exercise. Second, in this research, we focus on configurations that are driven by information sharing. A supply chain is certainly a value-creating network with multiple-channels of communication involving product, service and information (Parolini 1999). A possible area for future research is to bring product and service also into the scope of supply chain configurations. Third, this paper focuses on the methodology and the technology framework for configurable supply chains. A promising next step would be to test this framework through a real implementation and provide an empirical study in terms of its performance over typical supply chain management applications.

In addition, supply chain configurations must be further validated both syntactically and semantically. Syntactically, the ECA rules in each configuration must not conflict. Semantically, each parameter in an ECA rule should be checked. For example, if the “condition” clause in an ECA rule specifies the required order fill rate, this parameter can only be adjusted in a reasonable value range. In general, to ensure that a configuration is semantically correct, we may need to specify valid value ranges for each parameter. Therefore, a detailed procedure for validation should be developed. Finally, in the information sharing model, the cost of information sharing has been neglected. This assumption is reasonable since supply chain partners have highly automated enterprise systems and fast, integrated information exchange infrastructures. Thus, the cost differential of message exchange between a one message-per-week versus a one-message-per-day scenario is very small. On the other hand, the cost of information sharing may also need to consider the costs related to gathering data, preparing information in proper formats, processing shared information, maintaining data, and other data management efforts. Hence, a more accurate model should include this cost as another parameter.

7 Conclusions

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 new methodology to leverage information sharing for configuring supply chains based on well-known technologies, such as 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. Finally, we developed an architecture to implement our methodology and also discussed managerial implications of this work.

Acknowledgements The authors thank Steve Buckley, Dawn Russell, Alan Stenger, and John Yen for their helpful comments on an earlier version of this paper.

Reference

van der Aalst, W. M. P. (1999). Formalization and verification of event-driven process chains. *Information and Software Technology, 41*(10), 639–650.

Angulo, A., Nachtmann, H., & Waller, M. (2004). Supply chain information sharing in a vendor managed inventory partnership. *Journal of Business Logistics, 25*(1), 101–116.

Bensaou, M., & Venkatraman, N. (1995). Configurations of interorganizational relationshops: a comparison between U.S. and Japanese automakers. *Management Science, 41*(9), 1471–1492.

Chen, F. (2003). Information sharing and supply chain coordination. In A. G. de Kok & S. C. Graves (Eds.), *Handbooks in operations research and management science, Vol. 11, supply chain management: Design, coordination, and operation*. Amsterdam: Elsevier.

Chopra, S., & Meindl, P. (2001). *Supply chain management: Strategy, planning, and operations*. Upper Saddle River: Prentice Hall.

EDI ANSI X12. (1997). EDI V4 (004010), http://www.disa.org/bookstore/public/index.cfm

Finley, F., & Srikanth, S. (2005). 7 Imperatives for successful collaboration. *Supply Chain Management Review, 9*, 30–37.

Fox, M. S., Barbuceanu, M., & Teigen, R. (2000). Agent-oriented supply chain management. *The International Journal of Flexible Manufacturing Systems, 12*(2–3), 165–188.

Gosain, S., Malhotra, A., & El Sawy, O. A. (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*. Boston: Harvard Business School Press.

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

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

Kelton, D., Sadowski, R., & Sturrock, D. (2004). *Simulation with Arena*. 3rd edition, New York: Mc Graw Hill.

Lee, H. (2004). The triple-A supply chain. *Harvard Business Review, 82*, 102–112.

Li, Z., Kumar, A., & Lim, Y. (2002). Supply chain modelling—a co-ordination approach. *Integrated Manufacturing Systems, 13*(8), 551–561.

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

Liu, R., Kumar, A., & Aalst, W. (2007). A formal modeling approach for supply chain event management. *Decision Support Systems, 43*(3), 761–778.

Malhotra, A., Gosain, S., & El Sawy, O. A. (2005). Absorptive capacity configurations in supply chains: gearing for partner-enabled market knowledge creation. *MIS Quarterly, 29*(1), 145–187.

McCarthy, D. R., & Dayal, U. (1989). The architecture of an active database system. In J. Clifford, B. G. Lindsay & D. Maier (Eds.), *Proceedings of ACM SIGMOD conference on management of data* (pp. 215–224). New York: ACM Press.

Miller, H. (1996). The multiple dimensions of information quality. *Information System Management, 13*(2), 79–82.

NØkkentved, C., & Hedaa, L. (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).

Parolini, C. (1999). *The value net: A tool for competitive strategy.* England: Wiley.

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., Guo, Y., & Ganeshan, R. (1996). Inventory control under gamma demand and random lead time. *Journal of Business Logistics, 17*(1), 291–304.

Voluntary Interindustry Commerce Standards (VICS) Association. (2002). Collaborative Planning, Forecasting and Replenishment (CPFR) Version 2.0, http://www.vics.org/docs/committees/cpfr/CPFR_Tabs_061802.pdf.

Waller, M., Johnson, M. E., & Davis, T. (1999). Vendor managed inventory in the retail supply chain. *Journal of Business Logistics, 20*(1), 183–203.

XQuery. (2007). XQuery 1.0: An XML Query Language. W3C. Available at http://www.w3.org/TR/2007/REC-xquery-20070123/.

Rong Liu is a Research Scientist at IBM T. J. Watson Research Center. She received her Ph.D. in Business Administration from The Pennsylvania State University. Her research interest includes business process modeling, workflow systems, supply chain management, and model-driven business transformation.

Akhil Kumar is a professor of information systems at the Smeal College of Business at Penn State University. He received his Ph.D. from University of California, Berkeley, and has previously been on the faculties at Cornell University and University of Colorado, and also spent a sabbatical year as a scientist at Bell Labs, Murray Hill, NJ. He has done pioneering work in XML based workflows and data replication techniques. He has published more than 70 papers in academic journals and international conferences.