Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 1998 Proceedings

International Conference on Information Systems (ICIS)

December 1998

Applying Component Technology to Improve Global Supply Chain Network Management

Gek-Woo Tan
University of Illinois at Urbana-Champaign

Michael Shaw
University of Illinois at Urbana-Champaign

APPLYING COMPONENT TECHNOLOGY TO IMPROVE GLOBAL SUPPLY CHAIN NETWORK MANAGEMENT

Gek Woo Tan
Michael J. Shaw
Beckman Institute for Advances in Sciences and Technology
University of Illinois at Urbana-Champaign
U.S.A.

Abstract

A key requirement of supply chain management strategies is information sharing. Component technology facilitates information sharing by providing a means for integrating heterogeneous information systems into virtual information systems. The component environment makes possible new strategies to reshape and improve supply chain networks. These strategies include modularization and encapsulation, plug-and-play component development, enterprise specialization, dynamic supply chain network configuration, and cross-industry enterprise integration. We show how these strategies can be applied to the supply chain network management using the order fulfillment process as an illustration.

Keywords: Simulation and modeling IS, electronic commerce, supply chain management, component technology.

1. INTRODUCTION

A supply chain is a network of business units and facilities that procure raw materials, transform them into intermediate goods and then final products, and deliver the products to customers through a distribution system. Supply chain management (SCM) is concerned with the management of these activities such that the product passes through the chain in the shortest time with the lowest cost (Lee and Billington 1995). As supply chain networks (SCNs) become increasingly global, coordination between processes are more crucial. The focus of SCM has shifted from engineering efficient manufacturing processes to the coordination of activities in the SCN through knowledge management.

Information sharing is a key ingredient in coordination. Programs like the continuous replenishment program (CRP), just-in-time (JIT), and quick response programs rely on the dissemination of scheduling, shipment, or manufacturing information to the parties involved. Information sharing improves coordination between the supply chain processes and reduces inventory costs (Strader, Lin and Shaw 1998).

A major barrier to information sharing is the incompatibility of different computer systems. Frequently, each company develops its own proprietary systems which make information systems (IS) integration across enterprise boundaries difficult. Channel partners have to invest in electronic data interchange (EDI) technology in order to implement CRP or JIT strategies. The drawbacks of EDI are (1) a heavy investment for each participant and (2) the reach of EDI systems is limited because each EDI system is different due to a lack of standardization. Component technology overcomes the system incompatibility problem by encapsulating enterprise systems as object components, made accessible by standardized interfaces, and defining a protocol for transmitting documents between these components. Using component technology, we can form virtual information systems

(VIS) from different enterprises' heterogeneous IS. This improves SCN management by (1) reducing production costs through lower procurement and distribution costs, (2) better utilization of resources through enterprise specialization, and (3) greater integration of SCN activities through the virtual integration of IS.

This paper is organized as follows: we introduce the component technology and the strategies it facilitates in section 2. We illustrate how the order fulfillment process is improved in a component environment in section 3. We report the current project status in section 4 and conclude in section 5.

2. COMPONENT TECHNOLOGY

The concept of component technology was first developed for more productive software development. Component-based development (CBD) enables prefabricated, pre-tested, and reusable pieces of software to be assembled, thereby enabling very flexible applications to be built rapidly.

We define the component environment to comprise the following: software components (also called component objects), interfaces, component framework, broker, repository, and communication network. Szyperski and Pfister (1997) define a software component to be a unit of composition with contractually specified interfaces and explicit context dependencies. It can be independently deployed and is subject to composition by third parties. Software application programs, system management facilities, and other services are examples of component objects. A component's interface defines access points that allow its clients to access the services it provides. It is a set of methods that does not change throughout its lifetime and is globally uniquely identifiable. The interface may be hard-wired into the component (like COM objects) or separately implemented as interface adapters (like CORBA objects). A component framework is a software entity that supports components conforming to certain standards and allows instances of these components to be "plugged" into the framework (Szyperski 1998, p. 280). It establishes environmental conditions for the component instances and regulates how they interact with each other. The broker and repository are necessary for the successful implementation of a component-based software system. Given a request, the broker identifies a component instance that fulfills the request, locates it, and plugs it into the framework. Its function is to select and assemble component instances belonging to different IS into a VIS, analogous to the formation of virtual enterprises from separate enterprises. In current component technology software like CORBA and COM, the broker's function is limited to locating and forwarding required documents; it does not as yet possess the sophistication to perform selection and evaluation. The communication network is the network that physically links IS together.

Component technology provides seamless communications between applications residing in different IS. When a client object invokes a call, the request is passed to its interface adapter stub, which invokes the call to the broker. The broker searches from the repository for the location of the object that implements the request, passes to it the parameters, invokes its method, and returns the results to the client's interface adapter. The process is transparent to the client, who views the call as a local call (Orfali, Harkey and Edwards 1996, p. 68).

Figure 1 shows the creation of a VIS during the procurement process in a component environment. Assume that the *Manufacturer* wants to procure a part. The Broker looks for a part catalog from its repository, finds one in *Supplier*, and forwards it the request. The *Supplier* downloads its electronic catalog software to

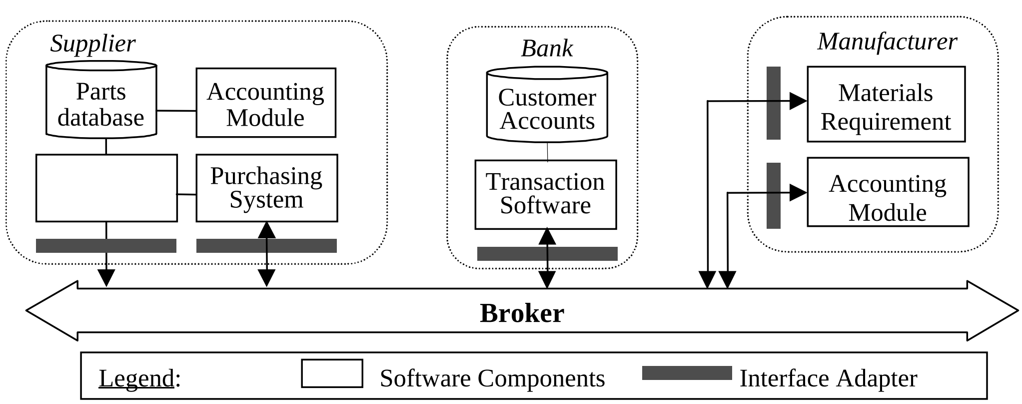


Figure 1. Virtual Information System in a Component Environment

the *Manufacturer's* IS to determine its requirements. The catalog software forwards the order to the *Supplier's* purchasing system, which processes the order and alerts the *Bank* about the transaction. The *Bank* plugs its transaction software into the accounting modules of both the *Manufacturer* and the *Supplier* to process the transaction. This appears to be a local process to both the *Manufacturer* and *Supplier*. At the end of the transaction, the *Supplier* and the *Bank* remove their software from their partners' IS. In this example, we see the formation of a VIS from the *Manufacturer, Supplier*, and *Bank.* Furthermore, the VIS grows and shrinks according to the needs of the enterprises.

Component technology makes possible strategies for information sharing and integration as follows:

1. *Modularization and encapsulation*—Each component in an IS can be separately developed or purchased, and then assembled to form the final system. The CBD process is (1) simpler because the inter-dependencies between components are reduced, (2) faster because components can be bought or developed by a third party, and (3) more flexible as components are acquired when needed.

2. *Plug-and-play component development*—CBD aims to develop component objects that can be easily plugged into frameworks with minimal user effort in integration. For example, a supplier who won a bid to supply to a manufacturer in an online bidding process can plug its procurement software to the manufacturer's IS to access its production schedule and requirements and automatically forward the orders to the supplier's IS.

3. *Enterprise specialization*—Because IS components can be combined, each enterprise can specialize in developing modules for its core processes and collaborate with enterprises specializing in complementary processes. For example, a retailer can turn over its procurement process to its supplier (as in CRP) or engage an independent third party vendor to perform the procurement process and focus its resources to better understand and meet consumer needs.

4. *Dynamic SCN configuration*—Specialization and collaboration go hand-in-hand. An enterprise that specializes in certain processes needs other enterprises to perform the other processes. Component technology enables enterprise partners to collaborate and assemble VIS according to their needs, resulting in dynamic SCN configuration.

5. *Integration of business processes of cross-industry partners*—One of the emergent behaviors in electronic commerce is in the convergence of services in different industries, especially in consumer services. For example, Web travel sites now allow consumers to make air, accommodation, and car rental reservations and constructs personalized travel itineraries. They contain links to information, such as weather forecasts and information about destination cities, that are of interest to travelers. In addition, they also serve as information dissemination centers where airlines and hotels inform consumers about their various promotions. The value offered by Web travel enterprises is to bring together the service providers from different industries and offer them to consumers in the form of a customizable bundle.

3. THE SUPPLY CHAIN NETWORK IN A COMPONENT ENVIRONMENT

We illustrate how the strategies in section 2 can be used to improve the SCN management. An SCN is composed of enterprises held together for the purpose of fulfilling the demand of the consumers and delivering the intermediate product from one enterprise to another as it travels through the process chain. An SCN can be viewed as a container with the enterprises as the components. In particular, the order fulfillment process (OFP) forms the framework that holds together the procurement process, order processing and forecasting, and distribution process.

The OFP is the heart of the SCN. The objective of SCN is to optimize the OFP by coordinating the activities and streamlining the transition of intermediate products from enterprise to enterprise via efficient logistics management. The OFP starts with the consolidation of demand forecasting and incoming orders. Based on the production quantity, the production schedule is generated and input materials procured. The enterprise then manufactures and assembles the products before shipping them to the downstream customer.

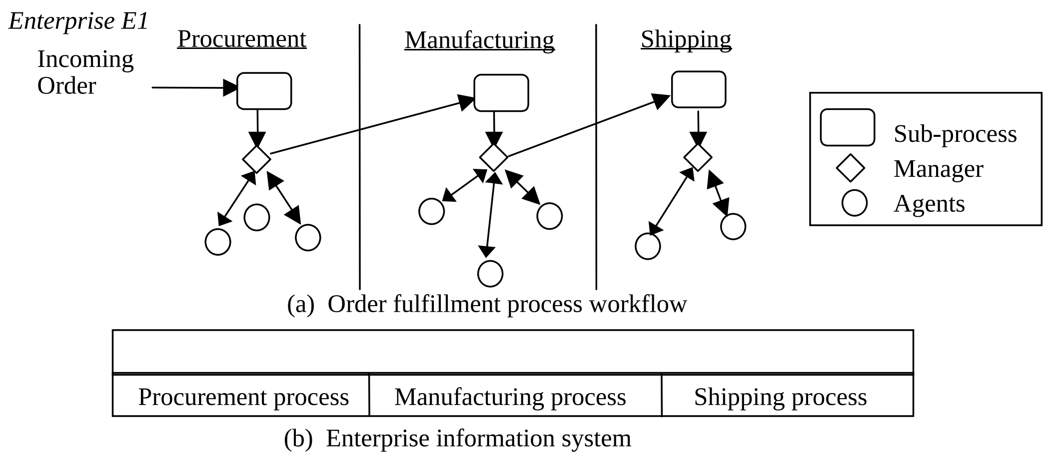


Figure 2. Order Fulfillment Process in a Single Enterprise

In a typical enterprise, all processes are owned and performed by the same enterprise *E1*, as shown in Figure 2a. The manager controls the agents who perform the various activities. The white icons indicate that all agents belong to *E1*. The IS caters to all the processes (see Figure 2b).

Figure 3a shows the OFP in a component environment where different activities in the OFP are performed by different enterprises. In the manufacturing process, for example, some parts are manufactured in-house (white icon) while other parts are outsourced or purchased from other enterprises (patterned icons). Some processes, like distribution, may be completely managed by an external enterprise. For some processes, coordination duties are shared by both *E1* and its partners. For example, staple input materials suppliers, like *S1*, can run *E1*'s scheduling software to find out its production schedule and deliver the input materials accordingly (see Figure 3b). The procurement manager just needs to handle one-time transactions and other unexpected procurements (i.e., *S2* and *S3*.) It negotiates short-term purchasing contracts with *S2* and *S3* via the online bidding service. The manufacturing manager places orders for the outsourced parts with manufacturers *M1* and *M2* through the EDI systems.

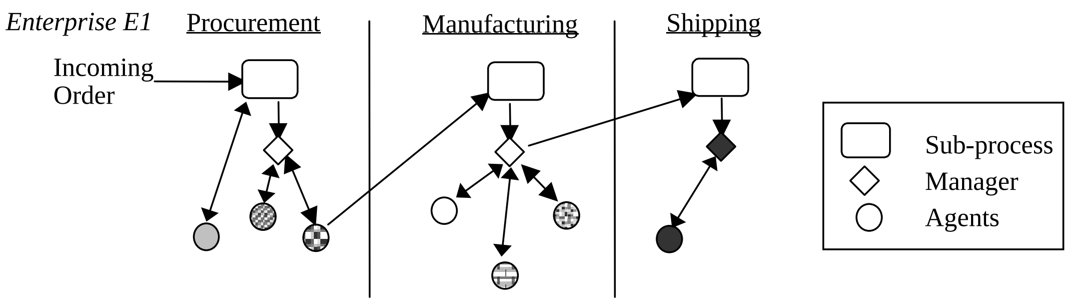
The IS in Figure 3b is component-based where each process's applications have been divided into modules. Some partners like *S1* can access *E1*'s scheduling software via the Broker while others like *D1* plugs its software directly plug into *E1's* IS. A VIS with varying degrees of integration and accessibility is formed between *E1*, *S1* and *D1*.

As a result, the OFP in Figure 3 is a multi-enterprise system with different tasks performed by different enterprises. Through component technology, an enterprise can share information with its distributor without having to develop an explicit EDI system. Through the Extranet, suppliers access the enterprise's scheduling software to obtain up-to-date production information and coordinate their output with the enterprise's production schedule. Hence, OFP becomes a collaborative effort instead of a single-enterprise effort. The participants of this SCN can also change dynamically. For example, in one-time only transactions, the procurement manager may select different suppliers each time.

4. CURRENT PROJECT STATUS

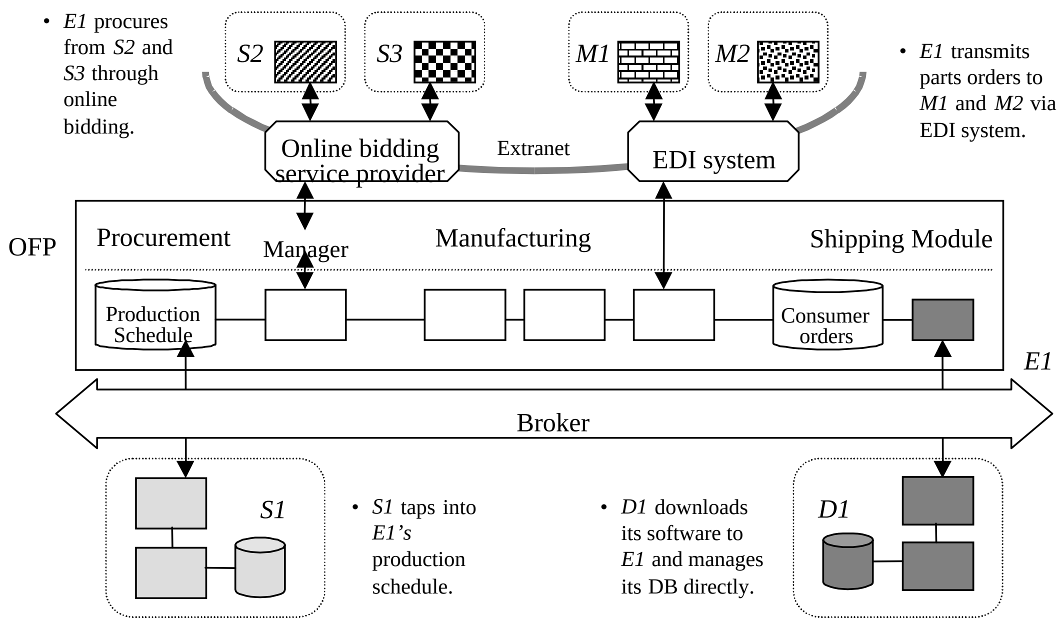
We chose multi-agent simulation as the platform for developing our model. Since simulation allows us to specify individual agent behavior and observe the interactions of the agents, it is a particularly suitable tool for demonstrating the adaptive behavior of SCNs. It also provides a testbed for testing and validating our hypotheses.

We developed a multi-agent simulation model to test different demand management policies as well as the impact of information sharing in the order fulfillment process (Lin 1996, Lin, Tan and Shaw 1998). Our findings confirm that information sharing improves the performance of the SCN. Information sharing, when combined with assemble-to-order policy, reduces inventory costs and shortens cycle time.



(White icons—own enterprise agents   Patterned icons—other enterprise agents)

(a) Order fulfillment process workflow



(b) Component-based information system

Figure 3. The Multi-purpose OFP in a Component Environment

We will develop a second simulation model to show that information technologies, in particular component technology and Extranet, improve SCN management. In this paper, we have shown how component technology facilitates information sharing through the integration of heterogeneous IS in the form of VIS. We will show how Extranet technology facilitates information exchange between an enterprise and its business partners as well as its customers through the Internet in a forthcoming paper. The simulation model will be used to verify three sets of hypotheses: first, information sharing reduces the bull-whip effect; second, the new strategies reduce the tradeoff between cost-efficient and market-responsive supply chain networks; third, the strategies result in new, emergent behaviors by the SCN.

Our models are developed using the Swarm toolkit, a simulation platform designed to simulate concurrent, distributed artificial world (Burkhart 1994). We chose Swarm because it is an object-oriented, dynamically scheduled, discrete-event simulator which makes it a natural platform for developing complex adaptive systems. Swarm is written in Objective-C and was developed at the Santa Fe Institute. It supports both batch simulations and interactive simulations and uses the Tcl scripting language and

Tk widget set to define its graphical interface. The current version of Swarm supports a variety of platforms including Windows95 and Windows NT.

5. CONCLUSION

This paper describes how component technology integrates the IS and strategies of enterprises to improve SCN management. Our first simulation program models the impact of information sharing in the OFP and confirms that information sharing improves the performance of the SCN. We hope to develop a second simulation model that incorporates the SCM strategies into our model by the end of this year.

References

Burkhart, R. “The Swarm Multi-agent Simulation System,” position paper for OOPSLA’94 at http://www.santafe.edu/projects/swarm/oopsla94.html, accessed on 4/29/1998.

Lee, H. L., and Billington, C. “The Evolution of Supply-Chain-Management Models and Practice at Hewlett-Packard,” *Interface* (25:5), September-October 1995, pp. 42-63.

Lin, F. *Reengineering the Order Fulfillment Process in Supply Chain Networks: A Multiagent Information System Approach*, unpublished Ph.D. thesis, University of Illinois at Urbana-Champaign, July 1996.

Lin, F.; Tan, G. W.; and Shaw, M. J. “Multi-Agent Enterprise Modeling,” *Journal of Organizational Computing & Electronic Commerce*, 1998, forthcoming.

Orfali, R.; Harkey D.; and Edwards, J. *The Essential Distributed Objects Survival Guide*, John Wiley & Sons Inc., New York, 1996.

Strader, T.;Lin, F.; and Shaw, M. J. “Simulation of Order Fulfillment in Divergent Assembly Supply Chains,” *Journal of Artificial Societies and Social Simulation* (1:2), 1998 (http://www.soc.surrey.ac.uk/JASSS/1/2/5.html).

The Santa Fe Institute. Swarm Web pages at http://www.santafe.edu/projects/swarm/, accessed on 11/12/1997.

Szyperski, C. *Component Software: Beyond Object-oriented Programming*, Addison-Wesley, Reading, MA, 1998.

Szyperski, C., and Pfister, C. “Workshop on Component-Oriented Programming,” in M. Mühlhäuser (ed.), *Special Issues in Objected-Oriented Programming—ECOOP96 Workshop Reader*, Verlag, Heidelberg, 1997.