

Optimizing Supply Chains through Service-Oriented Architecture

Xin Liu, Zhijun Zhang, Tingjie Lu and Wei Fan

School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing 100876, P.R. China liuxin1919@gmail.com
zhangzhj@bjtelecom.net lvtingjie@bbn.cn fanwei@buptinfo.com

Abstract. Supply chain is defined as a system of suppliers, producers, distributors, retailers and customers where material, financial and information flows connect participants in the directions. Most supply chains are composed of independent agents with individual preferences. It is expected that no single agent has the power to optimize the supply chain. Supply chain management is now seen as a governing element in strategy and as an effective way of creating value for customers. Service Oriented Architecture (SOA) is a systems architectural tool that support mission-critical transactions and improve systems integration for enterprises. A discussion on how to optimize enterprise's supply chains by the entire development service-oriented architectures, to centralize information, and to increased information flows, reduced uncertainty, and a more profitable supply chains. In the paper, Modeling Dynamic Supply Chain Management (MDSCM) system is put forward, it allows a producer to select, dynamically from the MDSCM Registry, suppliers of components, based on the price, availability, and delivery time of those components. A producer can use one of several strategies to aggregate customers' orders before it processes them and to accumulate suppliers' quotes before it decides on a particular supplier. The use of a Service Oriented Architecture, such as MDSCM, can substantially improve the efficiency of a supply chain.

Keywords: *Supply chain management (SCM), Service-oriented architecture (SOA), Service management, Supply chain, Electronic business (E-business)*

1. INTRODUCTION

Supply chain is defined as a system of suppliers, producers, distributors, retailers and customers where material, financial and information flows connect participants in the direction.

Service Oriented Architecture (SOA) has been viewed as a strategic approach to IT that provides increased flexibility. While the benefit of SOA in terms of organizational agility is well accepted, managers are increasingly becoming concerned about the net performance impact of SOA adoption. Applications communicate with each other in such architectures through services. Services are self describing components, which can be recognized by client applications through look up from a registry. The client application and the service provider communicate via

Please use the following format when citing this chapter:

Liu, X., Zhang, Z., Lu, T., Fan, W., 2007, in IFIP International Federation for Information Processing, Volume 254, Research and Practical Issues of Enterprise Information Systems II Volume 1, eds. L. Xu, Tjoa A., Chaudhry S. (Boston: Springer), pp. 405-414.

standard protocols and exchange information using standard data formats. SOA has come to prominence as previous software architectures base on object-oriented approaches suffer from a lack of standards when compared to SOA [1].

Business partners in a supply chain compose a business community, in which a set of enterprises shares a common market sector, holds existing mutual business relationships, has heterogeneous roles, sizes and technology, and has coordination with coordination support offered by some business authority. In order to satisfy requirements of partners in a supply chain each business community must meet following requirements. First, availability, security, extensibility as quality of services is the main key factors for service distribution in heterogeneous environments [2]. Second, interoperability, dynamic behavior, and many-to-many approach provide flexibility for supply chains [3]. Third, loose coupling, technical independence, and openness provide cost-effective software development. The SOA approach, used in the software development as an integration paradigm, is suggested as a solution to software complexity [4]. Service Oriented Architecture has been paid an increasing attention in both academic and industrial communities since software systems based-on it adapt well to the continuous changes of requirements and application environments.

In this paper, we propose the model of using SOA in the construction of innovative and supply chain management of an enterprise. The paper starts with an introduction of the SOA and then gives a description of supply chain management.

Supply chain management benefits from a variety of concepts that were developed in several different disciplines as marketing, information systems, economics, system dynamics, logistics, operations management, and operations research. There are many concepts and strategies applied in designing and managing supply chains; In giving modeling of sharing information in supply chains, the proposed model is a discrete dynamic model and the cooperation of units is based on contracts and formal agreements achieved in negotiation process; in modeling of sharing information in industry chain, the proposed model is a discrete dynamic model and the cooperation of units is based on protocols and formal agreements achieved in negotiation process; In giving a model of sharing services, the proposed model is a sharing service can be represented by four components.

SOA is a new paradigm in distributed systems aiming at building loosely-coupled systems that are extendible, flexible and fit well with existing legacy systems. SOA will be able to offer solutions that are both cost-efficient and flexible. An increasing number of companies in the world subscribe to the idea that using SOA can significantly improve the efficiency of supply chains and provide a way to ensure competitive advantage. In this paper, we propose to investigate the feasibility of using SOA in the construction of innovative and advanced Supply chains. The paper starts with an introduction of the Supply Chain and then gives a description of the Service - Oriented Architecture: its definition and functions, advantages; In section 4, the paper gives an introduction of Modeling Dynamic Supply Chain Management: Supply chain by physical layer of MDSCM and Information flow model between producers and customers; in section 5, MDSCM Architecture for supply chain is presented thereafter.

2. SUPPLY CHAIN MANAGEMENT

Supply chain management has generated a substantial amount of interest both among managers and researchers. Supply chain management is now seen as a governing element in strategy and as an effective way of creating value for customers. A structure of supply chains is composed from potential suppliers, producers, distributors, retailers and customers etc. The units are interconnected by material, financial, information and decisional flows. Most supply chains are composed of independent units with individual preferences. Each unit will attempt to optimize his own preference. Behavior that is locally efficient can be inefficient from a global point of view. In supply chain behavior is much inefficiency. An increasing number of companies in the world subscribe to the idea that developing long-term coordination and cooperation can significantly improve the efficiency of supply chains and provide a way to ensure competitive advantage.

The overall business environment is becoming increasingly dynamic. Demand and supply for custom products can be very dynamic. Supply chains operate in a network environment. Dynamic information and decision-making models are called to accommodate these changes and uncertainties.

There are some approaches to model and analyze the supply chain dynamics. Dynamic models of supply chains try to reflect changes in real or simulated time and take into account that the network model components are constantly evolving. An information asymmetry is a source of inefficiency in supply. Information exchange is a very important issue for coordinating actions of units. The expected result is a mutually beneficial, win-win partnership that creates a synergistic supply chain in which the entire chain is more effective than the sum of its individual parts. Supply chain partnership leads to increased information flows, reduced uncertainty, and a more profitable supply chain. The ultimate customer will receive a higher quality, cost-effective product in a shorter amount of time.

Supply chains involve identifying a market for offering, creating the offering, supplying products to customers, getting paid for them, managing the customer relationship, and repeating the process. Over the last decade, with the emergence of the Internet, the World Wide Web, related technologies, and globalization has affected how businesses interact with other businesses [5]. In this new environment, companies are seeking ways to become more flexible and adaptive in response to the competitive international economic environment.

A supply chain moves a product or service from the supplier to the customer. It typically consists of suppliers providing raw materials or services, producers putting different components together to produce products, warehouses storing raw materials and manufactured goods, distributors providing finished goods or services to customers, and customers purchasing goods or materials. The main objective of supply chain management is to achieve the most efficient use of resources to meet the customers' demands [6].

Typically, supply chain management deals with three types of flows:

1. Product flow - Movement of goods from a supplier to a customer as well as customer returns
2. Information flow - Transmitting orders and updating the status of delivery

3. Financial flow - Credit terms, payments, payment schedules, consignment, and title ownership

Supply chain management benefits from a variety of concepts that were developed in several different disciplines as marketing, information systems, economics, system dynamics, logistics, operations management, and operations research. There are many concepts and strategies applied in designing and managing supply chains.

3. SERVICE ORIENTED ARCHITECTURE

3.1 SOA's Definition and Functions

There are currently many definitions of the Service Oriented Architecture which are rather divergent and confusing. We choose to adopt the definition inspired by Sayed Hashimi [7]:

In SOA, software applications are built on basic components called services. Processes, etc., defined in terms of what it does, typically carrying out a business-level operation.

A service in SOA is an exposed piece of functionality with three properties:

1. The interface contract to the service is platform independent.
2. The service can be dynamically located and invoked.
3. The service is self-contained. That is, the service maintains its own state.

There are basically three functions that must be supported in a service-oriented architecture:

1. Describe and Publish service;
2. Discover a service;
3. Consume/interact with a service.

3.2 SOA's Advantages

There are many shades of meaning when you start digging into the details of what someone means when they use the term. We only describe SOA's advantages, Marks, Eric A. (2004) describe SOA's advantages [8]:

The SOA's key advantage is flexibility. Unlike previous paradigms such as client-server and mainframe environments, it offers IT functionality as cross-platform shared services. This offers a number of benefits, but the most immediate is a clear ROI arising from asset reuse. Once a portfolio of Web services is available in an SOA, these reuse benefits multiply exponentially in an "SOA network effect." The value of SOA increases with the number of available services, and the number of different applications or users that access those services. This value compounds over time and becomes even greater if SOA is leveraged both internally and externally.

SOA's flexibility also benefits an organization through faster application development and lowered costs by allowing hardware and software components to be reused.

Applications developed this way can even be of higher quality than those developed independently because the components are retested and the Web services interfaces have already been proven.

SOA's flexibility is particularly important when trying to integrate several different systems, such as those that result from mergers. After organic growth and multiple acquisitions, for example, office supply retailer Staples found itself with five duplicate systems for credit card authorizations. Rather than transition to just one authorization system or create multiple interfaces to each back-end application, Staples implemented a Web service for use by all five. An SOA provides the run-time, management, and security functionality, allowing Staples to route each of its 100 million annual credit card transactions through whichever one of the five banks and clearinghouses provides the best value for that particular payment.

SOA refers to any system that exposes its functionality as services. The next question, naturally, is "what's a service?" We will turn to a metaphor to explain that one: think of services as a mechanical watch with hands, numbers, and an internal mechanism. The hands and numbers are the "interface," and the mechanism is the "code." To do more than simply tell time—to function as a stopwatch, for example—a watch would need additional components, such as mechanisms to start and stop the time, to display the elapsed time, and to reset the timer. Those operations are essentially simple services.

SOA has led to the creation of many related terms, some that were created because of SOA and others that were given new life. The term loosely coupled, for example, refers to a property of systems in which the complexity of the system is partitioned inside a small number of building blocks that are connected in clearly defined ways. Loose coupling means that the building blocks do not depend on each other in complex ways and can easily be rearranged to meet new challenges. The idea of the service grid has also gained a lot of currency. A service grid is an infrastructure of many different services all designed to work together.

4. THE INTRODUCTION OF MDSCM

4.1 Supply Chain by Physical Layer of MDSCM

MDSCM is appropriate to proposed modeling framework for dynamic multilevel supply network.

The MDSCM language consists of four basic building and one space-saving tool. The four building blocks are:

- Stock, represent something that accumulates;
- Flow, activity that change magnitude of stock;
- Converter modifies an activity;

Connector transmits inputs and information.

Figure1 shows the icons of blocks. This approach enables to model and to solve a broad class of dynamic problems.

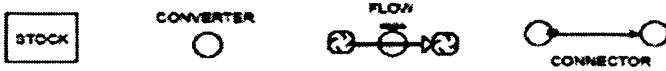


Figure 1. Block of the MDSCM Language

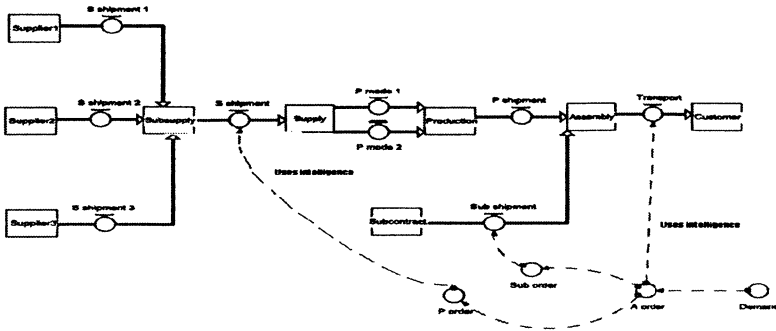


Figure 2. Supply Chain by Physical Layer of MDSCM

In MDSCM system, physical layer contains many functions that can facilitate dynamic modeling of supply chain (Figure2).

In Figure2, the intelligence (information) from producer’s clients is producer’s business secret, it don’t let producer’s rivals to acquire, so in the paper, intelligence represents producer’s business information from clients.

4.2 Information Flow Model Between Producers and Customers

In the context of the MDSCM defined in Section 3, the usage of our modeling framework is illuminated in Section 4.1. Since the services are elicited from business requirements, their related service components can be specified. Then, for architectural reuse and interoperability, the system architecture should be defined. The architecture specification should be given. Hereinto, it involves service component searching problem. And our before work on SE4SC [9] can support the query of matched service components in the repository.

With respect of the high dynamic nature, dynamic configuration and reconfiguration of SOA according to the changing requirements and environments at runtime and evolving as execution processes are supported here. SOA allows its services consumer to replace services at run-time when there are some new services, which is a better alternative to the former one concerning functionality or quality. And our method can support the dynamic configuration and reconfiguration of SOA

by remove or modifying the usage dependency between different service components. Besides, graph transformation rules referred in [10] may be helpful, too.

We can refine the architecture specification defined in figure 3. We suppose that there are an alternative retailer service component and an alternative producer service component in the MDSCM. On account of that the instance of the retailer service component are running on the unsteady thread in the Internet, it may collapse or stop without notification. In these cases, in order to satisfy the needs of customer all the time, it may require the instance to restart or request a substitute. Otherwise, when the published services provided by a service component have higher quality and reliability, consumer will be notified and it may rebind the new service component. According to above discussion, the refined architecture in the MDSCM is defined in Figure3.

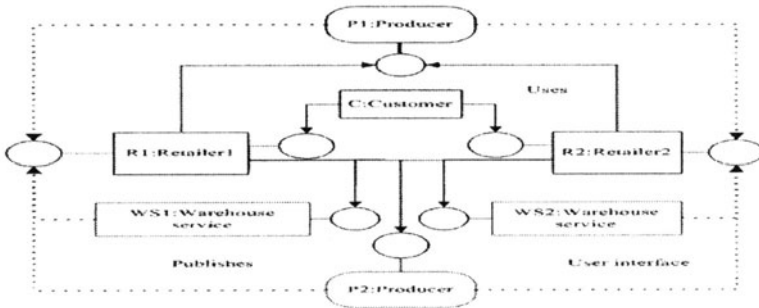


Figure 3. Refined Architecture about Information Flow Model in MDSCM

In Figure3, the information from producer’s clients is business secret, it don’t let producer’s rivals to acquire, because it don’t let producer’s rivals to acquire, it is called intelligence. The information that producers publish to their clients is not very important to the rivals.

5. MDSCM ARCHITECTURE

MDSCM provides a dynamic environment for customers, producers, and suppliers to cooperate as they have never done before. The MDSCM services in one enterprise dynamically interact with the MDSCM services in other enterprises, which are dynamically accessed over the World Wide Web. MDSCM supports communication between the producer and suppliers, even if the producer did not have any prior business with them.

The advantages of MDSCM for supply chains that it increases business flexibility and lets businesses adapt more quickly to changing business needs. Moreover, it enables applications to be composed in a loosely-coupled fashion, and allows software services to be reused because it has been designed with modularity in mind. A Service Oriented Architecture is fronted by a client user interface, and end users see only the user interface. Although the MDSCM client user interface that we have

developed is specific to computer manufactured from components obtained from different suppliers. The underlying MDSCM architecture is general and can be reused by producers and suppliers of other kinds of products.

Although we consider only a three-level supply chain and a single producer here, the MDSCM strategy generalizes to deeper supply chains with N levels, $N \geq 3$, as shown in Figure 4, where a producer is a supplier of the products it produces and a supplier is a producer of the supplies it offers. Note that MDSCM is present at the enterprises in the supply chain that act as both producer and supplier. By considering the entire supply chain, MDSCM provides a better understanding of supply chain needs and faster adaptation to changing demand and supply.

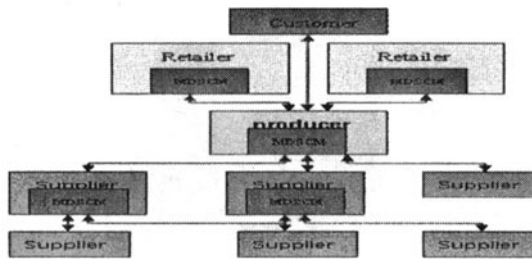


Figure 4. Use of MDSCM in the Supply Chain

The MDSCM system at the producer consists of the following components: the Materials Manager, the Orders Manager, the Database (DB) Monitor, the Registry (Directory Service), and the Quotes Manager. These components are shown in Figure 5.

Customers obtain information, from the Materials Manager, about the materials that are available. According to the information in the Orders Database, the Materials Manager relates a material to its components and relates a component to supplies at one or more suppliers. Receiving orders from customers, the Materials Manager passes the information to the Orders Manager. The Orders Manager inserts into the Orders Database information about the customer and products, that the customer is interested in purchasing, and manages the status of the components of each customer's product order. On receiving an order, the Orders Manager informs the DB Monitor to scan the orders. The DB Monitor checks the Orders Database and decides, depending on the particular strategy chosen, whether to inform the Quotes Manager to initiate a search for suppliers and to communicate with them.

All of these components play a role in the different phases of an order from a customer. There are two phases in processing a customer's order:

1. Waiting phase, which involves the collection of orders from the customers before making a Quote request for a component from the suppliers
2. Quotes phase, which involves the collection of Quote replies from the suppliers, and deciding on which supplier will provide the component.

The MDSCM architecture has interfaces on the customer side and the producer side. Each interface involves different components of MDSCM, and some components of MDSCM serve as a bridge between these two interfaces.

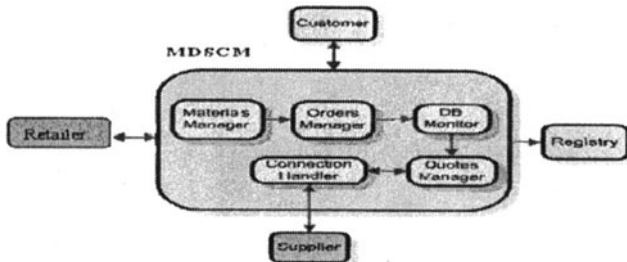


Figure 5. The Components of MDSCM

6. CONCLUSIONS

We have presented the MDSCM system, a Service Oriented Architecture for dynamic supply chain management. We have designed and implemented the services of MDSCM related to information flow. The MDSCM architecture as presented here deals with business processes up to the point where the decision to do business with a specific supplier is completed. However, the status of an order might change after the producer has placed an order with a supplier.

Supply chains need interoperability of business partners and business software and have specified business process and workflow management. Partners also agree their mutual relationships in MDSCM. Business process transformation to information system is done using a workflow notation with transactional properties and compensation. The implementation phase of the composite service uses coordination protocols, negotiation between partners, and WS-specifications.

Moreover, relationships and dependencies between the components that constitute a product might exist. We augment MDSCM with a work flow component that handles the relationships and dependencies between the components of a product.

Supply chains need interoperability of business partners and business software and have specified business process and workflow management. Partners also agree their mutual relationships in MDSCM. Business process transformation to information system is done using a workflow notation with transactional properties and compensation. The implementation phase of the composite service uses coordination protocols, negotiation between partners, and WS-specifications.

In this article we have shown how important it is to analyze business processes and design proper workflows based on the SOA. The proposed MDSCM pattern clarifies the most important concepts needed in the composite service implementation in the supply chain.

The performance of the MDSCM Registry that allows the producer to discover suppliers dynamically needs to be analyzed in terms of the overhead of the Quote messages and the synchronization messages. We expect that this overhead will affect the overall performance of the Query phase. If the Quotes Manager has all of the information it needs to make a decision, it can skip the Query phase and initiate purchases from the suppliers immediately.

REFERENCES

1. B. Lim and J. Wen, Web services: An Analysis of the Technology, Its Benefits, and Implementation Difficulties, *Information System Management*. Volume 20, Number 2, pp.49, (2003).
2. P. Baglietto, M. Maresca, A. Parodi, and N.I. Zingir, Deployment of Service Oriented Architecture for a Business Community, in *Proc. of the Sixth International Enterprise Distributed Object Computing Conference (EDOC'02)* (Lausanne, Switzerland, 2002), pp.293-304.
3. J. Bosch, *Design and Use of Software Architectures, Adopting and Evolving a Product-Line Approach* (Addison-Wesley: New York, NY, USA, 2000).
4. D. Linticum, *10 Things to Think about When Building Perfect SOA*, SOA Web Service Journal (2005). http://webservices.sys-con.com/read/121940_p.htm (Accessed December 2, 2006).
5. R. Murch, *Autonomic Computing* (Prentice Hall, Englewood Cliffs, NJ, USA, 2004).
6. Wikipedia, Supply Chain (2006). http://en.wikipedia.org/wiki/supply_chain (Accessed December 4, 2006).
7. A.E. Marks, Build a Better Enterprise Application, *Network Magazine*. Volume 19, Number 8, pp.18-24, (2004).
8. S. Hashimi, *Service-Oriented Architecture Explained*, O'Reilly ONDotnet.com (2003). http://www.ondotnet.com/pub/a/dotnet/2003/08/18/soa_explained.html (Accessed December 3, 2006).
9. H. Chen, S. Ying, J. Liu, and W. Wang, SE4SC: a Specific Search Engine for Software Components, in *Proc. 4th International Conf. on Computer and Information Technology* (Wu Han, China, 2004), pp.863-868.
10. L. Baresi, R. Heckel, S. Thöne, and D. Varró, Modeling and Analysis of Architectural Styles based on Graph Transformation, in *Proc. 25th Int. Conf. Software Engineering Workshop on CBSE6: Automated Reasoning and Prediction* (Portland, 2003), pp.67-72.