The MIDAS System:
A Service Oriented Architecture for Automated Supply Chain Management

Firat Kart, Zhongnan Shen, Cagdas Evren Gerede
Department of Electrical and Computer Engineering
Department of Computer Science
University of California, Santa Barbara, CA 93106
fkart@ece.ucsb.edu, szn@cs.ucsb.edu, gerede@cs.ucsb.edu

Abstract

The MIDAS system that we have developed is an automated supply chain management system based on the Service Oriented Architecture and Web Services. MIDAS provides a loosely-coupled distributed environment that allows customers, manufacturers, and suppliers to cooperate over the Internet and World Wide Web. It aims to reduce inventory carrying costs and logistics administration costs, yielding a more efficient supply chain, by supporting just-in-time manufacturing. It eliminates, or substantially reduces, human intervention on both the customer/manufacturer side and the manufacturer/supplier side. It allows a manufacturer to select, dynamically from the MIDAS Registry, suppliers of components, based on the price, availability, and delivery time of those components. A manufacturer 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 MIDAS, can substantially improve the efficiency of a supply chain.

1 Introduction

Since the earliest days of business, the fundamental business processes haven't changed. They 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, certain factors have led to a major shift in business management thinking, which has had a significant impact on how companies do business. New business models have emerged with the Internet, the World Wide Web, and related technologies, such as the Service Oriented Architecture (SOA) [5, 12, 13] and Web Services [1, 16, 20], and globalization has affected how businesses interact with other businesses [11].

In this new environment, competition is intense; financial pressures are unrelenting; and changes in customer demands, technological innovations, and government regulations are continuous and often sudden and dramatic. Companies are seeking ways to become more flexible and adaptive in response to the competitive international economic environment.

Supply chains are profoundly challenged by this new 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, manufacturers 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 [14].

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

- *Product flow* - Movement of goods from a supplier to a customer as well as customer returns
- *Information flow* - Transmitting orders and updating the status of delivery
- *Financial flow* - Credit terms, payments, payment schedules, consignment, and title ownership

The MIDAS (Managing Integrated Demand and Supply) system that we have designed and implemented focuses on the management of information flow. It has the goal of leveraging existing IT infrastructure to enable users to automate their supply chains. MIDAS is based on the Service Oriented Architecture [5, 12, 13] and employs Web Services [1, 16, 20] to achieve its objective of automated supply chain management.

As a Service Oriented Architecture, MIDAS ensures that the IT systems of the different enterprises can adapt quickly and easily to support rapidly changing business needs. MIDAS supports horizontal business processes that are distributed across multiple enterprises. Through its use of Web Services, it provides interoperability between legacy back-end enterprise software systems. The MIDAS software is modular, which allows it to be reused at multiple levels of the supply chain.

MIDAS provides a Registry that allows a manufacturer to discover the existence of suppliers, to select suppliers dynamically, and to find an alternate supplier on demand, if an existing supplier becomes unavailable, and redirect its requests seamlessly to that alternate supplier. At the manufacturer, MIDAS receives orders from the manufacturer's customers, and places orders with the manufacturer's suppliers, automatically and dynamically. MIDAS allows a manufacturer and its suppliers to negotiate a business deal on-line by either accepting a quote as is, or modifying it. MIDAS makes it easier for small suppliers to get into business with larger manufacturers by automating the procurement process. Most importantly, MIDAS aims to meet the needs of the customers on time and to eliminate the need for a large inventory at the manufacturer and thus reduce costs.

A live demonstration of the MIDAS supply chain management system is available at http://phoenix.ece.ucsb.edu/SupplyChain/.

2 The Business Model

The business model for the MIDAS system is based on data obtained from the Council of Supply Chain Management Professionals and the Census Bureau of the U.S. Department of Commerce. According to a report of the Council of Supply Chain Management Professionals [15], the business logistics costs of the United States in 2004 were $1,015 Billion. These business logistics costs include inventory carrying costs, transportation costs, shipper related costs, and logistics administration costs. Business logistics costs data for 2005 and 2006 are not yet available, but they were projected to continue to increase. Inventory carrying costs represent a considerable portion of the total logistics costs, and include the costs of interest, taxes, obsolescence, depreciation, insurance, and warehousing. According to the U.S. Department of Commerce, Census Bureau, the total value of all business inventory has increased dramatically to more than $1.26 Trillion by the end of 2004.

By using an automated supply chain management system, these business logistics costs, particularly the inventory carrying costs and the logistics administration costs, can be reduced. The MIDAS system is an automated supply chain management system based on the Service Oriented Architecture and Web Services that aims to reduce these costs, resulting in a more efficient supply chain.

The particular manufacturing model employed affects the overall performance of an automated supply chain management system. Using the just-in-time manufacturing model, Dell [2] exhibits the current best practice of supply chain management with no inventory or almost no inventory. Because computer technology improves so rapidly, computers and their components depreciate very quickly and, thus, cutting inventory is a financial necessity. Bill Breen, senior projects editor at Fast Company, reports that Dell considers inventory as fish: *The more you keep it the faster it deteriorates.* This phenomenon is true not only for high technology products, like computers, but also for low technology products, such as apparel. However, even Dell can do better in automating the supply chain process. To the best of our knowledge, Dell still employs human beings using email, faxes, and telephone calls to place orders for computer components with its suppliers [6].

By applying SOA practices to supply chain management, the MIDAS system that we have developed aims to eliminate or reduce substantially human intervention and, thus, reduces costs, yielding a more efficient supply chain.

3 The MIDAS Architecture

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

The advantages of MIDAS for supply chain applications, and of the Service Oriented Architecture in general, are 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 MIDAS client user interface that we have developed is specific to computers manufactured from components obtained from different suppliers. The underlying MIDAS architecture is general and can be reused by manufacturers and suppliers of other kinds of products.

Although we consider only a three-level supply chain and a single manufacturer here, the MIDAS strategy generalizes to deeper supply chains with N levels, $N \geq 3$, as shown in Figure 1, where a manufacturer is a supplier of the products it produces and a supplier is a manufacturer of the supplies it offers. Note that MIDAS is present at the enterprises in the supply chain that act as both manufacturer and supplier. By considering the entire supply chain, MIDAS provides a better understanding of supply chain needs and faster adaptation to changing demand and supply.

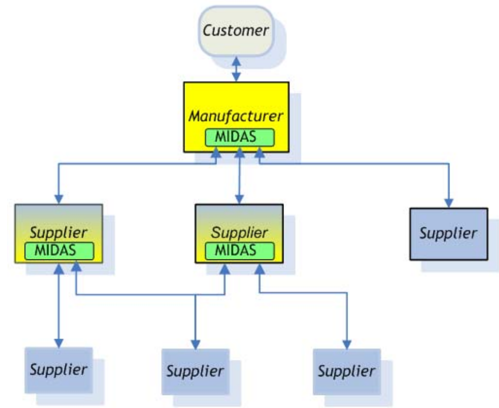

Figure 1. Use of MIDAS in the supply chain

The MIDAS system at the manufacturer 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 2.

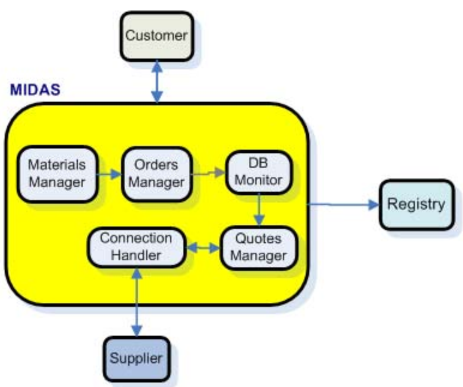

Figure 2. The components of MIDAS

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. The strategies used by the DB Monitor to decide whether or not to inform the Quotes Manager are discussed in Sections 3.2 and 4.1. The *Quotes Manager* handles communication messages and relates Quote requests and Quote replies. The strategies used by the Quotes Manager to decide whether to stop collecting Quote replies and to make a decision on a supplier are discussed in Sections 3.2 and 4.2.

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:

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

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

3.1 The Customer Side

The customer side of MIDAS deals with the customer/manufacturer communication and interaction. Inspired by Dell's build-to-order model, it enables the customers to customize their orders using an order service that communicates with other services to obtain the components necessary to manufacture the product and also to arrange shipping and financing. In this way, customers are able to build their own customized products.

The MIDAS architecture is based on the premise that it is the customer's opinion that counts. MIDAS enables a customer to customize the products that the customer purchases by choosing the materials that constitute those products. A *material* consists of one or more components that are required to make the product. A *component* is a supply that can be obtained from one of several suppliers. For example, a computer consists of components that include the processor, memory chip, graphics card, network interface card, etc. For each component, there are one or more suppliers from which the manufacturer can obtain that component, and the manufacturer can select dynamically the supplier of that component.

With MIDAS, customers are able to save their customized product orders for as long as two weeks before they decide to purchase. After receiving an order, the manufacturer gathers the components together from the suppliers in order to produce the product. Customers are informed about the status of not only the material, but also the components that constitute the material.

3.2 The Manufacturer Side

By applying SOA practices to supply chain applications, we aim to automate the supply chain and, thereby, reduce human intervention, errors, and costs. The customer/manufacturer interface receives information from the customer and displays the status of the customer's orders. The manufacturer/supplier interface provides the following functionality:

- Monitors orders
- Searches for suppliers
- Contacts related suppliers
- Decides on the best supplier(s) from which to obtain components

The MIDAS system aims to meet the needs of the customers on time and to eliminate the need for a large inventory at the manufacturer.

3.2.1 The Orders Manager

MIDAS introduces the concept of *logical inventory* into supply chain management. Logical inventory is data, stored in the supply chain management system's computers and databases and related to the collected customers' demands and the customers' orders. With logical inventory, the manufacturer does not need to maintain a large physical inventory in its warehouses.

MIDAS assumes that the more orders the manufacturer accumulates the more gain it has. Purchasing thousands of items has a different per item price than the per item price of a single item. For this reason, the Orders Manager tries to accumulate as many orders for a particular component as it can. Thus, the Orders Manager collects orders for products from the customers and aggregates several instances of the components of the products for those customer orders.

An *order event* is triggered when more than a specific number of items (instances of a component) are requested, a timeout occurs, a maximum waiting time is reached, or some combination of these criteria occurs. In contrast to physical inventory, a decision point is reached, and the processing of an order can start when one of these criteria is satisfied.

3.2.2 The Database Monitor

To keep track of orders, MIDAS employs a Database (DB) Monitor that scans the Orders Database for each product that might lead to the triggering of an order event.

When the DB Monitor triggers an order event, it informs the Quotes Manager to find appropriate suppliers and communicate with them. The main problem for manufacturers that depend on their suppliers is that, if their suppliers are not available, the manufacturer's supply chain system cannot progress. In a dynamic environment, the manufacturer must be able to find new suppliers on demand and to satisfy the customers' needs in a timely manner.

The DB Monitor makes a decision to inform the Quotes Manager using one of several strategies:

- System user decides when to place orders
- Timeout occurs at which an order needs to be placed
- Threshold number of items from the different customer orders is reached
- Combination of the timeout and the threshold number of items
- Average waiting time for a set of orders is reached

When the DB Monitor informs the Quotes Manager, the Quotes Manager initiates communication with the suppliers of the component, requesting the aggregated number of items.

The DB Monitor retrieves information (supply ID, amount to order, average waiting time) about the components from the Orders Database. According to the strategy used, the DB Monitor decides whether a set of orders satisfies the quote to be requested.

3.2.3 The Registry

The Registry at the manufacturer keeps information about the suppliers, namely, the contact information for the suppliers and the products that they have to offer. A supplier registers itself and the products it has to offer in the Registry. The manufacturer and the suppliers are assumed to use the same supply IDs, and the supply IDs are assumed to identify, uniquely, the supplies across the manufacturer and the different suppliers. Figure 3 shows the Registry and the communication links and protocols, aswell as the structure of the Registry, that MIDAS uses.

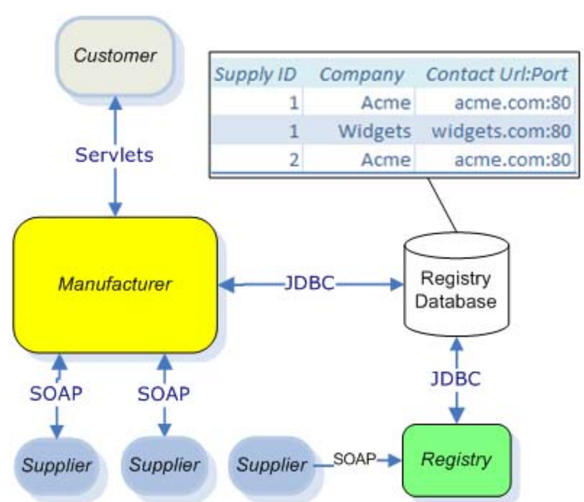

Figure 3. Communication in MIDAS

In MIDAS, the Registry includes the contact information of each supplier and the supply IDs of the products that it offers. This choice assumes that the suppliers will not frequently update the supplies that they offer. If there are thousands of suppliers listed in the Registry, and it contains only contact information, the cost of sending Quote requests to all suppliers would be too high, particularly if there are only a few Quote replies from the suppliers. MIDAS still uses synchronization messages to update the Registry with the suppliers' contact information and supply IDs, but there are fewer of them. By not keeping price information in the Registry, MIDAS allows the manufacturer to negotiate the price with the supplier at which it is willing to buy. The supplier might accept the offered price or revise the quote and send a reply with an updated price.

First, the supplier registers by communicating with the Registry at the manufacturer and passing current information about itself to the Registry. The supplier provides its contact information and the products that it sells. If there is a change (such as running its applications at a different URL or port, or offering new supplies), it is the supplier's responsibility to inform the Registry. After registering, the supplier can receive Quote requests from the manufacturer. The Registry enables the manufacturer to find relevant suppliers of the components it needs on demand.

3.2.4 The Quotes Manager

Having decided on the suppliers with which it will communicate, the Quotes Manager at the manufacturer sends quote requests to the suppliers initiating the second phase, the Quotes phase. After sending the quote requests, the Quotes Manager can use one of several strategies:

- System user decides
- Wait for quotes from all suppliers and decide
- Wait for quotes from all suppliers until some number of replies is received
- Wait for quotes from all suppliers until a specific time
- Hybrid of time and number of items threshold
- Average waiting time for supplies

In MIDAS, the Quotes Manager at the manufacturer asks for quote information from the suppliers, related to the proposed price, the number of items, and the proposed delivery time. However, the manufacturer can define the number of items needed, the type of supply, and the delivery time expected. Based on the suppliers' offers, the manufacturer can then update its Quote request and send the updated request to the supplier again.

Figure 4 shows an example sequence diagram for the Quotes phase. In the example, the Quotes Manager could not communicate successfully with Supplier C, perhaps because of a communication failure or unavailability of the suppliers' service. However, the Quotes Manager is aware of the total number of quotes that it sent and the number of Quote replies that it received. The Quotes Manager triggers events to make a decision on the suppliers with which it will do business. The decision point depends on the strategy used, and could be before the Quotes Manager receives Quote replies from all of the relevant suppliers. Having decided on a quote, the Quotes Manager then updates the status of the customer sale associated with the quote and informs the Orders Manager.

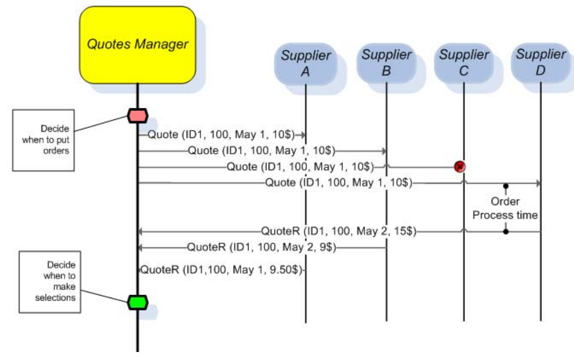

Figure 4. Quote requests and Quote replies

Thus, there are two important decision points, shown in Figure 4, that affect the time the customer has to wait to obtain information about a component's price and delivery time. The Orders Manager tries to accumulate as many orders for a particular component as it can, while not increasing the average customer waiting time. Similarly, the Quotes Manager tries to make the delay to the customer as short as possible, while not missing better quotes that might appear after it has made a decision.

4 Evaluation

We have discussed different strategies that the DB Manager can use in the Waiting phase to aggregate orders from the customers before it triggers the Quotes Manager to communicate with the suppliers. We have also discussed different strategies that the Quotes Manager can use in the Quotes phase when it accumulates replies from the suppliers. The delay for aggregating orders, delay_{AO}, affects the delay for accumulating quotes, delay_{AQ}, particularly if the supplier processes Quote replies sequentially.

To evaluate the different strategies that the DB Manager and the Quotes Manager can use, we developed a simulation of the MIDAS supply chain system, in addition to our implementation. We considered two aspects: the customers' satisfaction level and the manufacturers' gain. The *customers' satisfaction level* is measured by the average customer response time, and the *manufacturers' gain* is measured by the number of orders aggregated in each Quote request or the best price ratio of orders.

4.1 The Waiting Phase

First, we evaluated three of the strategies that the DB Manager can use in the Waiting phase:

- Aggregating a specific number of orders
- Timeout for a specific amount of time
- Average waiting time for orders

For each of these strategies, we fixed the delay due to the accumulation of Quote replies by the Quotes Manager and used the number of orders aggregated to measure the gain of the manufacturer.

4.1.1 Aggregating a Specific Number of Orders

This strategy aggregates a specific number of orders before the Quotes Manager issues a Quote request to the suppliers. The experimental results are shown in Figures 5 and 6.

As we can see from Figure 5, the average customer response time increases with the number of orders aggregated. The customer response time consists of the waiting time for the order, delay$_{AO}$, and the quote time for the component, delay$_{AQ}$. Thus, the response time for the customer is delay$_{AO}$ + delay$_{AQ}$. In this experiment, the quote time delay, delay$_{AQ}$, is almost constant, because we used the strategy of obtaining replies from all of the suppliers. When more orders are aggregated, more orders wait for a longer time, which increases the waiting time for the orders.

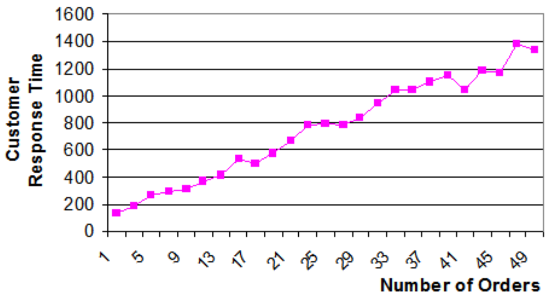

Figure 5. Relationship between the customer response time and orders accumulation

Figure 6 exhibits the conflict between customer satisfaction and manufacturer gain. Aggregating more orders benefits the manufacturer at the price of displeasing the customers. To keep the satisfaction of customers above a certain level, we can consult the graph to find the corresponding number of orders that can be aggregated.

4.1.2 Timeout for a Specific Amount of Time

The timeout strategy aggregates customer orders for a fixed amount of time and then the Quotes Manager issues a Quote request to the suppliers. The results for the timeout strategy are similar to those for the specific number of orders strategy. With a Poisson arrival rate, the number

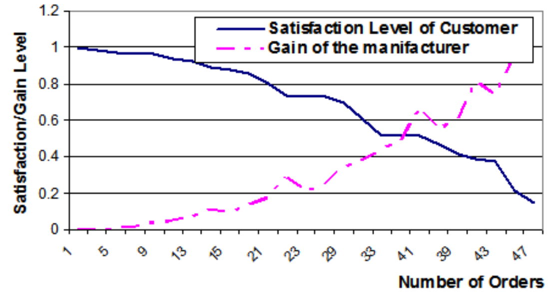

Figure 6. Comparison of customer satisfaction and manufacturer gain

of orders aggregated is proportional to the duration of the time period, which explains why the results for these two strategies are similar.

4.1.3 Average Waiting Time for Orders

In the average waiting time strategy, the DB Monitor scans the orders received and computes the average waiting time for the orders in the database. If the average waiting time exceeds some threshold, these orders are processed and the Query Manager then sends a Quote request to the suppliers.

Using a Poisson arrival process, again we obtain a similar curve for the customer response time. The reason is that, with Poisson arrivals, the longer the average waiting time is, the longer the delay$_{AO}$ is. Therefore, more orders are aggregated. However, the minimum customer response time for the average waiting time strategy is slightly smaller than that of the other strategies. If we choose a particular customer response time, the specific number of orders strategy outperforms the average waiting time strategy in terms of the gain of the manufacturer.

However, it is not always the case that the specific number of orders strategy is superior to the average waiting time strategy. For example, if the order arrival distribution is bursty with "spike" characteristics, the manufacturer is better off by adopting the average waiting time strategy, particularly when the average response time is low.

The specific number of orders strategy can use the long silent interval to aggregate orders to the specific number, which increases the delay$_{AO}$. However, the average waiting time strategy can always process the orders in time when the threshold is low because it takes the delay$_{AO}$ into account.

4.2 The Query Phase

Next, we evaluated two of the strategies for collecting replies that the Quotes Manager can use in the Query phase:

- Timeout for a specific amount of time
- Average waiting time for the replies

For each of these strategies, we used the average waiting time strategy in the Waiting phase and the best price ratio of orders to measure the gain of the manufacturer.

4.2.1 Timeout for a Specific Amount of Time

When the Quotes Manager collects replies from the suppliers, it waits for a specific amount of time, delay$_{AQ}$. It ignores the late replies and fills the customer orders immediately. Because it doesn't need to collect replies from all of the suppliers, this strategy reduces the customer response time compared to the strategy of collecting all replies. On the other hand, it cannot always obtain the best price from the suppliers, because the reply with the best price might come late and be ignored.

The customer response time and the best price ratio increase with the timeout value. If the system waits more time for the replies, which increases the delay$_{AQ}$, the customer response time (delay$_{AO}$ + delay$_{AQ}$) increases, and the probability that the best price is in a Quote reply increases. When the timeout value is close to 150, which is the maximum response time for the suppliers in our experiment, the best price ratio approaches 1.

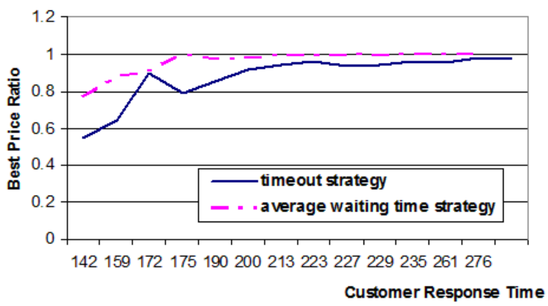

Figure 7. Comparison of timeout strategy and average waiting time strategy

4.2.2 Average Waiting Time for the Replies

In this strategy, if the average waiting time exceeds a certain threshold, the Quotes Manager stops collecting replies from the suppliers and completes the customer orders. Again, this reduces the customer response time compared to the strategy of collecting all replies, at the cost of reducing the best price ratio. The best price ratio for the average waiting time strategy approaches the maximum ratio quickly. Figure 7 shows that the average waiting time strategy outperforms the timeout strategy in terms of the best price ratio. For a fixed customer response time, the average waiting time strategy always has a higher best price ratio than the timeout strategy.

4.3 Discussion of Results

To enhance the satisfaction of the customers, for example, by reducing the customer response time, the system should process orders as soon as possible. On the other hand, aggregating orders can benefit the manufacturer, by reducing the manufacturer's costs. There is always a conflict between the customer and manufacturer, so a balance point must be found. According to the simulation results, if we fix the maximum customer response time, we can find the maximum number of orders that can be aggregated.

The strategies used in the Waiting phase have similar performance under the Poisson arrival distribution. The customer response time increases for each strategy evaluated. The specific number of orders strategy has slightly better performance than the average waiting time strategy in terms of the number of orders aggregated. However, the average waiting time strategy outperforms the specific number of orders strategy if the order arrival distribution is bursty, when the customer response time is relatively small.

The strategies used in the Query phase try to reduce the response time by controlling the number of Quote replies received. The best price from the suppliers might not be captured because not all of the Quote replies are taken into account. The timeout strategy and the average waiting time strategy in the Query phase reduce the customer response time at the cost of degrading the best price ratio. The average waiting time strategy approaches the maximum best price ratio more quickly than the other strategies, having better performance in terms of the best price ratio criterion.

5 Related Work

The MIDAS system for supply chain management aims to improve the efficiency and effectiveness of supply chains, in particular the customers' satisfaction level and the manufacturers' gain. Other researchers [4], who have a similar goal, base their business model on the four criteria of profit, lead time, performance, and promptness of delivery. They analyze supply chain performance at two levels, the chain level and the operation level. At the chain level, objectives associated with the criteria are set for each supply chain stage to satisfy customer service targets and to select the best supply chain management strategy. At the operation level, manufacturing and logistics activities are optimized for the given targets.

Other researchers [9] have investigated the use of UML for building a flexible supply chain business model. They regard a supply chain as five view models with four business domains, where each domain consists of functions, resources, processes, interactions, and business rules. However, they do not provide a system that illustrates their approach, as we do for MIDAS.

The MIDAS architecture, which is inspired by Dell's build-to-order model, enables customers to build their own customized products before they order. Other researchers [17, 19] have investigated customization in the supply chain. Companies other than Dell are now considering, or should be considering, the build-to-order model in order to achieve higher profitability.

In evaluating MIDAS we have considered the customers' satisfaction level, measured by the average customer response time, and the manufacturers' gain, measured by the number of orders aggregated in each Quote request or the best price ratio of orders. In contrast, other researchers [10] have considered the average supplier response time for evaluating the effectiveness of a supply chain architecture.

Other researchers [8] have considered the modeling and optimization of supply chain networks that involve multiple production sites and multiple suppliers, based on a mathematical programming problem that minimizes all costs in the network. Still other researchers [18] have considered optimized coordination of supply chains based on relationships and dependencies.

6 Conclusion and Future Work

We have presented the MIDAS system, a Service Oriented Architecture for automated supply chain management. We have designed and implemented the services of MIDAS related to information flow. In addition, we plan to design and implement the services related to product flow and financial flow.

The MIDAS 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 manufacturer has placed an order with a supplier. For example, if the delivery time changes, this information needs to be updated and the customer needs to be informed.

Moreover, relationships and dependencies between the components that constitute a product might exist. For example, a particular kind of video card might be usable only with a particular kind of motherboard; therefore, the video card cannot be processed before the motherboard is determined. We plan to augment MIDAS with a work flow component that handles the relationships and dependencies between the components of a product.

The performance of the MIDAS Registry that allows the manufacturer 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.

MIDAS might also use historical data to allow the manufacturer to make decisions at two decision points, the decision to issue Quote requests after collecting orders from the customers and the decision to stop collecting Quote replies and place an order with a particular supplier. Depending on the rate at which the customers place orders, the DB Monitor can adjust its timeout or threshold values accordingly. Moreover, the Quotes Manager can select better suppliers dynamically by waiting until it receives Quote replies from suppliers with good Quote replies that it previously ignored because they were late.

7 Acknowledgments

This research was supported in part by a UC Discovery Grant and QAD, Inc, grant number COM05-10194, and NSF grants IIS-0101134 and IIS-0415195. The authors wish to thank Professors Louise Moser, Michael Melliar-Smith, and Jianwen Su for their help with the project.

References

[1] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, "Web Services Concepts: Architectures and Applications," Springer-Verlag, Berlin, 2004.

[2] B. Breen, "Living in Dell Time," http://www.fastcompany.com/magazine/88/dell.html, referenced June 2006

[3] Computer World, "Supply Chain Management," http://www.computerworld.com/softwaretopics/erp/story/0,10801,66625,00.html, referenced June 2006.

[4] L. Dong and C. O'Brien, "Integrated Decision Modeling of Supply Chain Eficiency," *International Journal of Production Economics*, vol. 59, no. 1-3, March 1999, Elsevier, Netherlands, pp. 147-157.

[5] T. Erl, "Service-Oriented Architecture (SOA): Concepts, Technology, and Design," Prentice Hall, August 2005.

[6] T. Friedman, "Globalism is Good," http://www.impactlab.com/modules.php?name=News&file=article&sid=5338, referenced June 2006.

[7] General Motors, "History of General Motors," http://www.gm.com/company/corp_info/history/gmhis1920.html, referenced June 2006.

[8] D. Giglio and R. Minciardi, "Modelling and Optimization of Multi-Site Production Systems in Supply Chain Networks," *Proceedings of the 2003 IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2678-83.

[9] K. Jinho and K. J. Rogers, "An Object-Oriented Approach for Building a Flexible Supply Chain Model," *International Journal of Physical Distribution and Logistics Management*, vol. 35, no. 7, 2005, Emerald, UK, pp. 481-502.

[10] Y. S. Li, F. F. Ye, Z. M. Fang, and J. G. Yang, "Flexible Supply Chain Optimization and Its SRT Analysis," Industrial Engineering and Management, vol. 10, no. 1, February 2005, Springer-Verlag, China, pp. 89-93.

[11] R. Murch, "Autonomic Computing," On Demand Series, IBM Press.

[12] E. Newcomer and G. Lomow, "Understanding SOA with Web Services," Independent Technology Guides, December 2004.

[13] OASIS, "Reference Model for Service Oriented Architecture," http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm, referenced June 2006.

[14] Wikipedia, "Supply Chain," http://en.wikipedia.org/wiki/ supply_chain, referenced June 2006.

[15] R. Wilson, Council of Supply Chain Management Professionals, "16th Annual State of Logistics Report," June 2005.

[16] World Wide Web Consortium (W3C), "Web Services Architecture," http://www.w3.org/TR/ws-arch, referenced June 2006.

[17] J. Yang, S. Z. Zhao, and J. R. Wang, "Study on Postponement in Customization Supply Chain," *Industrial Engineering and Management*, vol. 10, no. 4, August 2005, pp. 35-44.

[18] T. Z. Zhao and Y. H. Jin, "Optimized Coordination of Supply Chain Based on Dependency," *Computer Integrated Manufacturing Systems*, vol. 10, no. 8, August 2005, CIMS, China, pp. 929-933.

[19] D. Zhou, B. H. Xiang, and G. S. Zou, "The Strategy of Mass Customization and the Countermeasure of Chinese Enterprises," *Industrial Engineering and Management*, vol. 8, no. 5, October 2003, pp. 12-16.

[20] O. Zimmerman, M. R. Tomlinson, and S. Peuser, "Perspectives on Web Services: Applying SOAP, WSDL and UDDI to Real-World Projects," Springer-Verlag, Berlin, 2003.