We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



14

Managing and Integrating Demand and Supply Using Web Services and the Service Oriented Architecture

Firat Kart, Louise E. Moser and P. M. Melliar-Smith
Department of Electrical and Computer Engineering
University of California, Santa Barbara
U.S.A.

1. Introduction

The Internet and the World Wide Web have had a significant impact on business management thinking and practice. Globalization has affected how businesses interact with other businesses, and even how divisions within companies interact with each other, and has increased competition and consolidation worldwide (Murch, 2004). The new business models are supported by modern information technology such as the Service Oriented Architecture (Erl, 2005; Moser & Melliar-Smith, 2008; Newcomer and Lomow, 2004; OASIS, 2006) and Web Services (Alonso *et al.*, 2004; Champion *et al.*, 2002; Chatterjee & Webber, 2003). Supply chains (ComputerWorld, 2006; Wilson, 2005) are particularly affected by these developments in business and technology.

The MIDAS (Managing and Integrating Demand and Supply) system that we have developed is an automated supply chain management system based on the Service Oriented Architecture and Web Services. The benefits of MIDAS (and of the Service Oriented Architecture in general) for supply chains are that it increases business flexibility and it enables businesses to adapt more quickly to changing business needs. The MIDAS system provides a loosely coupled distributed environment that allows customers, manufacturers, and suppliers to cooperate over the Internet and the World Wide Web. Generally, supply chain management considers three types of flow:

- **Information flow**, which pertains to placing, transmitting and filling orders, and updating their delivery status
- **Product flow**, which involves movement of goods from a supplier to a customer, as well as customer returns
- **Financial flow**, which relates to credit terms, payments, payment schedules, consignment, and title ownership.

The MIDAS supply chain system described here focuses, in particular, on the management of information flow. MIDAS, which is inspired by the build-to-order business model (Ghiassi & Spera, 2003; Gunasekarana & Ngai, 2005), enables customers to customize their products before they order. At the manufacturer, MIDAS receives orders from the customer,

Source: Supply Chain, The Way to Flat Organisation, Book edited by: Yanfang Huo and Fu Jia,
ISBN 978-953-7619-35-0, pp. 404, December 2008, I-Tech, Vienna, Austria

Open Access Database www.intechweb.org

and places orders with the suppliers, automatically and dynamically. The manufacturer can use one of several strategies to aggregate customers' orders before it starts processing them and to accumulate suppliers' quotes before it decides to do business with the suppliers. MIDAS allows manufacturers and suppliers to conduct a business deal online either by accepting a quote as is, or by negotiating.

MIDAS aims to reduce inventory carrying costs and logistics administration costs, yielding a more efficient supply chain, by supporting on-demand, just-in-time manufacturing. MIDAS leverages existing IT infrastructure to enable users to automate their supply chains. It makes it easier for small suppliers to get into business with large manufacturers by facilitating the procurement process. It reduces human intervention on both the customer/manufacturer side and the manufacturer/supplier side. Most importantly, MIDAS aims to meet the needs of the customers on time, and to reduce the costs of the manufacturer by eliminating the need for a large inventory.

In this article we describe the design and implementation of the MIDAS supply chain system. We discuss how MIDAS, and the Service Oriented Architecture and Web Services, support adaptation to change, interoperability, and scalability for supply chains, and how they can substantially improve the efficiency of a supply chain. We also present an evaluation of the MIDAS system, in terms of the customer's satisfaction as measured by the customer's response time, and the manufacturer's gain as measured by the number of orders aggregated or the best price ratio of orders.

2. The MIDAS architecture

As a Service Oriented Architecture (SOA), MIDAS ensures that the IT systems of different enterprises can adapt quickly and easily to support rapidly changing business needs. MIDAS supports horizontal business processes that are distributed across the Internet between, among and within enterprises. Through its use of Web Services, MIDAS provides interoperability between legacy back-end enterprise software systems.

By applying SOA practices to supply chain applications, MIDAS aims to automate the supply chain and, thereby, reduce human intervention, errors, and costs. The MIDAS software is modular, which allows it to be re-used at multiple levels of the supply chain. Moreover, as a SOA based on Web Services, MIDAS allows applications to be composed in a loosely-coupled fashion, and to be modified without disrupting the services provided to the customers.

MIDAS enables customers, manufacturers, and suppliers to cooperate in a dynamic environment, as they have never done before. The MIDAS system in one enterprise interacts dynamically with the MIDAS system in other enterprises, which are accessed over the Internet and the World Wide Web. MIDAS supports communication between the manufacturer and the suppliers, even if the manufacturer did not have any prior business with those suppliers and, thus, it increases the ease of collaboration between them.

MIDAS uses a UDDI Registry to allow a manufacturer to discover and select suppliers dynamically based on price, availability and delivery time of components. If an existing supplier becomes unavailable, it allows the manufacturer to find alternate suppliers on demand and to redirect its requests seamlessly to an alternate supplier. MIDAS uses a Reservation Protocol (Zhao *et al.*, 2008) to improve the performance of business transactions that span multiple enterprises in the supply chain. Use of the Reservation Protocol also decreases the probability of inconsistencies for business transactions between the manufacturer and suppliers.

Typically, a Service Oriented Architecture is fronted by a client user interface that uses the underlying services. Depending on the business application, a customized user interface is provided for customers that use the underlying services. Although the MIDAS client user interface that we have developed is specific to the manufacturing of a computer, the components of which are obtained from different suppliers, the underlying MIDAS system is general and can be used by manufacturers and suppliers of other kinds of products.

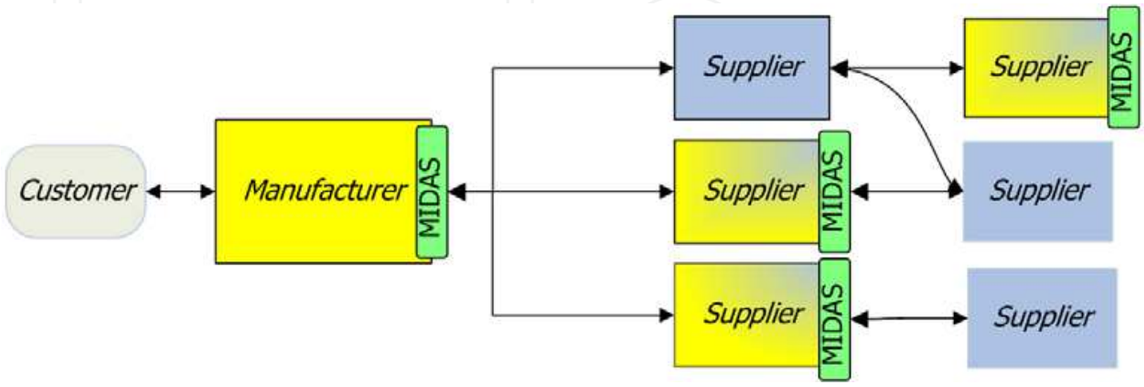


Fig. 1. Use of MIDAS in a supply chain

We consider here a three-level supply chain and a single manufacturer; however, as shown in Figure 1, the MIDAS strategy generalizes to deeper supply chains with N levels, $N \geq 3$, where a manufacturer is a supplier of the products it manufactures and a supplier is a manufacturer of the supplies that it offers. MIDAS is present at the businesses in the supply chain that act as both manufacturer and/or supplier. By considering the entire supply chain, MIDAS captures supply chain needs more effectively and provides faster adaptation to changing supply and demand.

MIDAS makes use of the following concepts in supply chain management. A *material* consists of one or more components. For example, a material for a computer is a particular computer model, such as Dell XPS 1310, Sony Vaio SZ780, *etc.* A *component* is a particular category for which there are one or more supplies. For example, the components of a computer include the processor, memory chip, graphics card, network interface card, *etc.* A *supply* is a product that is produced by a supplier. For example, the supplies associated with a processor might include a 2GHz CPU and a 3GHz CPU. A supply can be obtained from one or more suppliers, and the manufacturer can select the supplier of that supply dynamically. A *supply item* is an instance of a supply. For example, the supply items in a manufacturer's order from a particular supplier might include 100 2GHz CPUs.

MIDAS also uses the concept of *logical inventory* in supply chain management. Logical inventory is data, stored in the computers and databases of the supply chain management system that are related to the customers' needs and the customers' orders. With logical inventory, the manufacturer does not need to maintain a large physical inventory in its warehouses but, rather, can obtain the supplies that it needs on demand.

At the manufacturer, the MIDAS system comprises the following modules: Materials Manager, Orders Manager, Database (DB) Monitor, Communication Manager, and Quotes Manager. These modules are shown in Figure 2.

Customers obtain product information from a catalog provided by a Customer Web Service, which retrieves information from the Materials Manager. The Materials Manager relates a

material to its components and a component to its supplies. On receiving orders from the customers, the Materials Manager passes the information to the Orders Manager. The Orders Manager inserts, into the Orders Database, information about the customers and the products that the customers are interested in purchasing, and manages the status of the customers' orders. On receiving an order, the Orders Manager informs the DB Monitor, which scans the orders and triggers a business activity that starts purchasing supplies from suppliers. 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 communicate with them. The Quotes Manager handles Quote requests, and relates Quote replies to Quote requests.

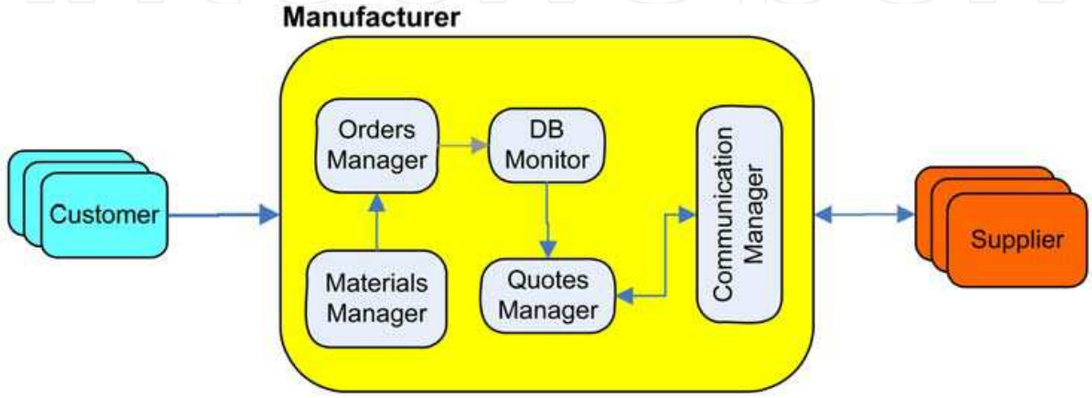


Fig. 2. The components of MIDAS

Each of these modules plays a role in the two phases of the manufacturers' processing an order from a customer. These two phases are:

- **Waiting phase**, which involves the collection of orders from the customers before making Quote requests for aggregated supply items from the suppliers
- **Quotes phase**, which involves the collection of Quote replies from the suppliers before making a decision on which supplier will provide the particular supply.

The strategies used by the DB Monitor in the Waiting phase to decide whether to stop collecting customer orders and to inform the Quotes Manager are discussed in Sections 2.2 and 4.2. The strategies used by the Quotes Manager in the Quotes phase to decide whether to stop collecting Quote replies and to make a decision on a supplier are discussed in Sections 2.2 and 4.3.

The MIDAS architecture has interfaces on the customer side and the manufacturer side. Each interface involves different modules of MIDAS, and some modules of MIDAS serve as a bridge between the two interfaces. The Customer Service and the Manufacturer Service are discussed below.

2.1 The Customer Service

The MIDAS architecture is based on the premise that it is the customer's opinion that counts. Inspired by Dell's build-to-order model, MIDAS enables a customer to customize a product that the customer purchases by choosing the materials that constitute that product. The customer customizes a product order using a Customer Web Service that communicates with other services to obtain the materials necessary to manufacture the product and to arrange shipping and financing of the product.

The customer side of MIDAS deals with customer/manufacturer communication and interactions. The Customer Service provides the following functionality:

- Provides catalog information
- Receives information from the customer
- Displays the status of the customer's orders.

The customer user interface uses the Customer Web Service to interact with MIDAS. Existing Web Services frameworks ease the process of implementing the user interface that interacts with the services. In Section 3 we discuss Web Services frameworks in more detail.

2.2 The Manufacturer Service

The customers send their orders for products to the manufacturer, and the manufacturer processes the orders. At the manufacturer, MIDAS does not contact the suppliers each time it receives an order from a customer. Rather, it uses one of several strategies (discussed below) to aggregate orders from different customers. However, the manufacturer must not take too long to confirm a customer's order with the price, product delivery time, *etc*.

The Manufacturer Service provides the following functionality:

- Monitors orders
- Searches for suppliers
- Contacts relevant suppliers
- Decides on the supplier(s) from which to obtain supplies.

This functionality is discussed below in terms of the Orders Manager, DB Monitor, Quotes Manager, and Registry, shown in Figure 2.

2.2.1 The Orders Manager

MIDAS assumes that, the more orders the manufacturer accumulates, the more gain the manufacturer has. Thousands of supply items have a per item price that is different than the price the price of a single item. The Orders Manager tries to accumulate as many orders for a particular supply as it can.

The Orders Manager collects orders from different customers and aggregates supply items (instances of a particular supply) for those customer orders. The Orders Manager then forwards the orders to the Database Monitor.

2.2.2 The Database Monitor and the Orders Database

The Database (DB) Monitor keeps track of the orders for each supply in the Orders Database and triggers order events, based on one of several strategies discussed below. When the DB Monitor triggers an order event for a particular supply, the processing of the order begins. The DB Monitor then informs the Quotes Manager to find appropriate suppliers for that supply and to communicate with them.

The main problem for manufacturers that depend on suppliers is that, if a supplier is not available, the manufacturer cannot make progress. In a dynamic environment, a manufacturer must be able to find new suppliers as the demand arises and to satisfy the customers' needs in a timely manner.

The DB Monitor uses one of the following strategies to trigger an order event and to initiate the processing of an order for a particular supply by informing the Quotes Manager to place an order for that supply.

- **System user decides when to place orders:**
  The DB Monitor provides information to the system user at the manufacturer about the number of supply items (instances of a particular supply) required for the order of that supply and the wait times for those items. The user triggers manually the Quotes Manager, which initiates communication with the appropriate suppliers.
- **Threshold number of supply items for a particular supply is reached:**
  On receiving orders from the customers, the Orders Manager directs the DB Monitor to scan the database. If the number of supply items is greater than the threshold, the DB Monitor informs the Quotes Manager.
- **Timeout occurs at which an order needs to be placed:**
  The DB Monitor scans the database at defined intervals. For each scan, when a timeout occurs, the DB Monitor informs the Quotes Manager.
- **Hybrid of threshold number of orders and timeout:**
  This strategy combines the threshold strategy and the timeout strategy, such that whichever occurs first triggers the Quotes Manager.
- **Average wait time for a set of orders is reached:**
  The Orders Manager accumulates the number of supply items for the orders from different customers. When a component has an average wait time that is greater than a threshold, the DB Monitor informs the Quotes Manager.

The DB Monitor retrieves information (supply ID, amount to order, average wait time, *etc*) from the Orders Database. According to the strategy used, the DB Monitor decides whether it is time to contact the suppliers of a supply and then informs the Quotes Manager to do so.

2.2.3 The Quotes Manager

The Quotes Manager initiates communication with the suppliers, requesting the aggregated number of supply items. It submits Quote requests to selected suppliers, receives Quote replies with a proposed price, number of items, and proposed delivery time, and then decides on a supplier with which to place an order.

The Quotes Manager retrieves information about the respective suppliers of a supply from the UDDI Registry. Having decided on the suppliers with which it will communicate, the Quotes Manager sends Quote requests to those suppliers initiating the second phase, the Quotes phase. After sending Quote requests, the Quotes Manager uses one of the following strategies to decide on a particular supplier with which to place an order. (It could order items for a particular supply from multiple suppliers but, for simplicity, we consider only a single supplier for that supply.)

- **System user decides on a supplier:**
  The system user has control over the processing of quotes from the suppliers and deciding on a supplier of a particular supply.
- **Threshold percentage of Quote replies:**
  The Quotes Manager continues to aggregate Quote replies until the total number of Quote replies is above a threshold percentage. For example, when the threshold percentage is 100%, the Quotes Manager waits until it receives Quote replies from all of the respective suppliers.
- **Timeout for Quote replies from the suppliers:**
  The Quotes Manager continues to wait until a specific time. If the Quotes Manager receives all of the expected Quote replies before the timeout, the Quotes Manager initiates processing of the Quote replies.

- **Hybrid of threshold number of Quote replies and timeout:**
  This strategy combines the strategies of waiting for a threshold number of Quote replies and waiting for a specific time, whichever occurs first.
- **Average wait time threshold for Quote replies:**
  The Quotes Manager accumulates Quote replies and when the average wait time for the Quote replies is greater than a threshold, the Quotes Manager initiates processing of the Quote replies.

The manufacturer can define the number of items needed, the type of supply, and the delivery time expected. Based on the suppliers' offers, the Quotes Manager then updates its Quote request and sends the updated Quote request to the suppliers again, continuing the negotiation with them.

Figure 3 shows an example sequence diagram for the Quotes phase. In this example, the Quotes Manager could not communicate successfully with Supplier C, perhaps because of a communication failure or unavailability of Supplier C's service. However, the Quotes Manager is aware of the total number of Quote requests that it sent and the number of Quote replies that it must receive. The Quotes Manager decides on the suppliers with which it will do business. Having decided on the status of a quote, the Quotes Manager updates the status of the customer order associated with the quote and informs the Orders Manager. Completion of a quote does not necessarily mean completion of a customer order. The customer order is completed once a decision about all of the different components for that order is made.

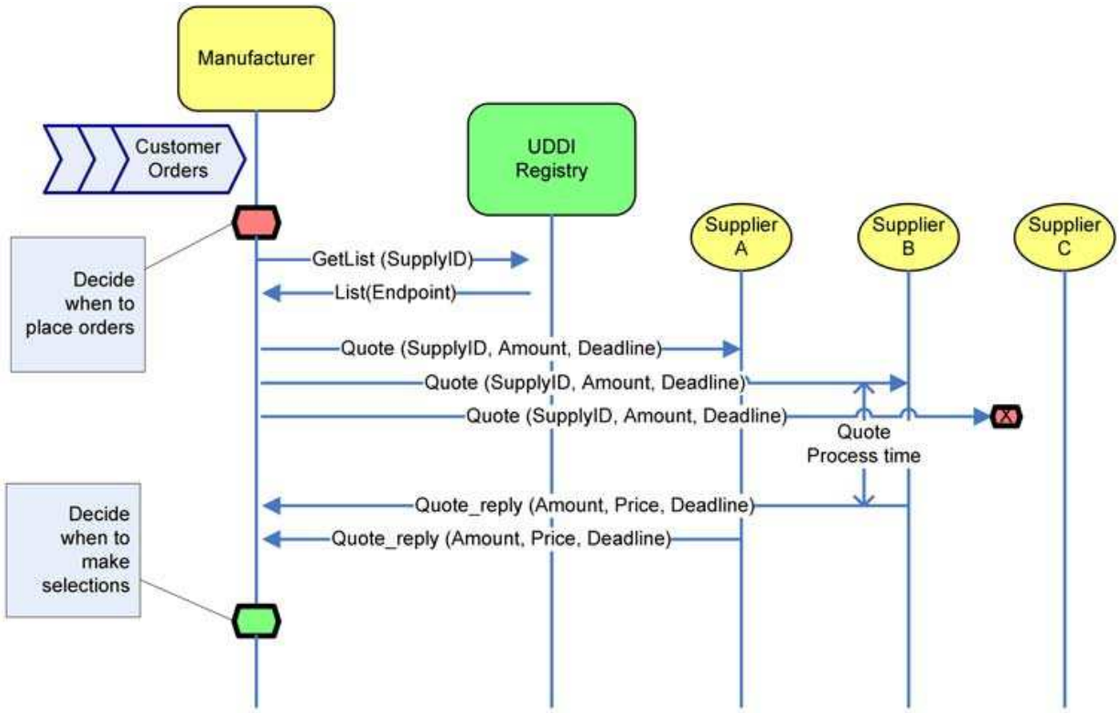


Fig. 3. Quote Request and Quote Reply interactions

As Figure 3 shows, there are two important decision points that affect the time the customer must wait to obtain information about the price and delivery time of the product. First, the Orders Manager accumulates orders for particular supplies, collecting as many as it can,

while not increasing the average customer waiting time, before it decides to proceed to the next decision. Next, the Quotes Manager accumulates Quote replies from the suppliers, aiming to make the delay for the customer as short as possible, while not missing better quotes after it decides.
Note that, while waiting for Quote replies, the Quotes Manager is not blocked. All of the messages are sent asynchronously, and no assumption is made about the order of delivery of Quote replies. The time at which the Quotes Manager receives Quote replies from the suppliers varies according to the supplier's quote processing time.

2.2.4 The UDDI Registry

The UDDI Registry enables the manufacturer to find relevant suppliers of the supplies it needs on demand. The Registry Database keeps information about the suppliers, in particular, the contact information of each supplier and the supply IDs of the products that it offers. 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 different manufacturers and different suppliers. The Registry supports a fair business environment for suppliers where they have a chance of doing business along with their competitors.

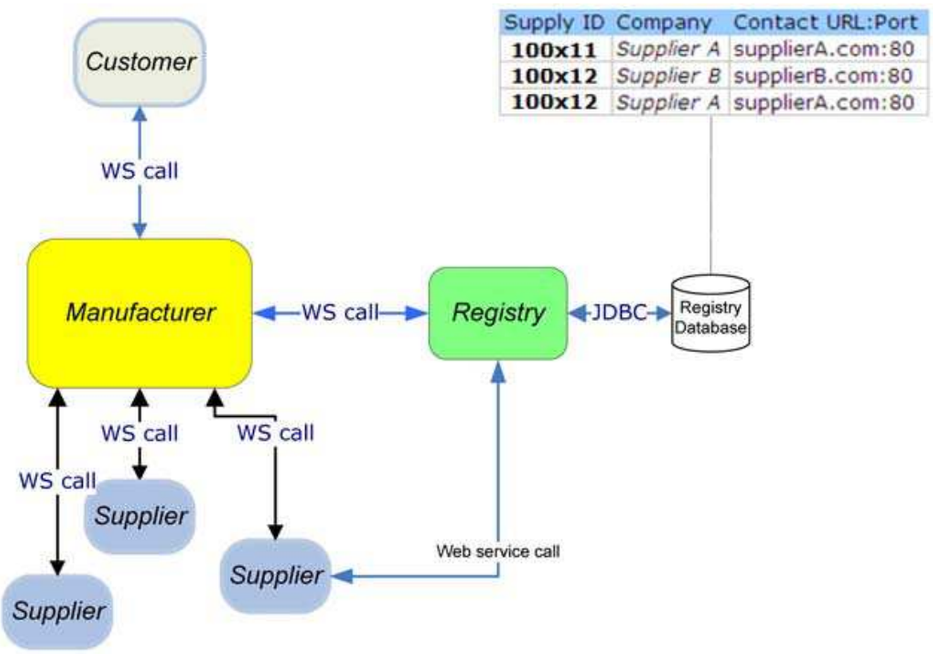


Fig. 4. Communication links and protocols used between the Manufacturer, the Registry, and the Suppliers

Figure 4 shows the Registry and the communication links and protocols, as well as the structure of the Registry. A supplier registers by communicating with the Registry and passing current information about itself to the Registry. The supplier provides its contact information and the products that it offers. If there is a change (such as a change of address, offering of new products, or removing product information), the supplier must update this information in the Registry. After registering, the supplier can receive Quote requests from manufacturers using MIDAS.

There are tradeoffs between the amount of information that is kept in the Registry and the number of Update messages sent by the suppliers. The two extreme strategies for the amount of information maintained by the Registry are:

- **Maintain only contact information for the suppliers:**
  In this case, the manufacturer must contact every supplier listed in the Registry, even though some suppliers do not provide what the manufacturer needs. The suppliers send Update messages (re register) containing only their contact information.
- **Maintain all the information about the suppliers:**
  Here, all the information about the suppliers includes contact information, particular supplies that they offer, maximum number of supply items that they can provide, price per item, projected delivery time, etc. The Registry simply returns a list of suppliers with their contact information sorted according to the criteria that the manufacturer wants to see.

In the first strategy, during the Quotes phase, the manufacturer spends considerable time sending and receiving messages in accumulating Quote replies from the suppliers.
In the second strategy, the suppliers frequently send Update messages to keep their information in the Registry up-to-date. However, during the Quotes phase, there are fewer messages, because the manufacturer needs to contact fewer suppliers; in particular, it can eliminate suppliers that have higher prices and later delivery times. Keeping the price and delivery time information in the Registry not only complicates the running of the Registry but also increases the amount of information that the Registry holds. Moreover, a supplier might not be willing to reveal all of its price information and the kind and number of supply items in the Registry.
Our implementation uses an intermediate strategy between these two extremes. The Registry includes only the contact information of the suppliers and the supply IDs of the supplies that they offer. This choice assumes that a supplier will not frequently update its catalog. If there were thousands of suppliers listed in the Registry, and the Registry contains only contact information, the cost of sending Quote requests to all of them would be quite high, particularly if there were only a few suppliers for a particular supply. Consequently, for the hybrid strategy, the Registry performs better in terms of message overhead than the alternative strategies. Moreover, by not keeping price information in the Registry, MIDAS allows the manufacturer to negotiate the price at which it is willing to buy from a supplier. The supplier might accept the offered price or revise its quote and send a Quote reply with an updated price. Update messages that update the Registry with the suppliers' contact information and supply IDs are still required, but there are fewer of them.

2.3 The Reservation Protocol

MIDAS employs the Reservation Protocol, which is an extended transaction protocol that is designed for business transactions that span multiple enterprises (Zhao *et al.*, 2008). Business activities between the manufacturer and the supplier are executed as two steps. The first step involves an explicit reservation of resources according to the business logic. The second step involves the confirmation or cancellation of the reservation. For example, a manufacturer that is interested in buying a product from a supplier sends Quote requests to the suppliers in the first step. Once the manufacturer receives Quote replies from the suppliers and makes a decision, the manufacturer sends confirmation or cancellation messages to the suppliers.

Alternative transaction methods suffer in terms of response time and throughput when there are concurrent requests. During the transaction of a request, resources are not available to other requests, increasing the response time and decreasing the throughput. Because the Reservation Protocol executes each step as a separate traditional short-running transaction, resources are classified as available and reserved, which differs from blocking the resources until the current request is complete. Use of the Reservation Protocol decreases the wait time of customers, because MIDAS can complete the business transactions between the manufacturer and a supplier much faster, even though there are many concurrent requests arriving at the suppliers.

The Reservation Protocol improves the performance of business transactions that span multiple enterprises in a multi-level supply chain. As shown in Figure 1, an enterprise can deploy both client and/or server middleware depending on business needs. A supplier at level N receiving a Quote request from a manufacturer at level N-1 can contact its suppliers at level N+1 to make a reservation to reply to a Quote request from a manufacturer at level N-1. Receiving Quote replies from the suppliers at level N+1, the supplier at level N can send a Quote reply to a manufacturer at level N-1. Depending on the decision of the manufacturer at level N-1, the supplier at level N sends a decision to its suppliers at level N+1. For traditional transaction protocols, the same scenario can result in longer delays and also inconsistencies in the databases of the different enterprises in the supply chain.

3. Implementation

The MIDAS supply chain system comprises software modules for the customers, the manufacturers, and the suppliers, as described in Section 2. The Web Services for MIDAS are built on the Apache Axis2 Framework (which is the core engine for Web Services built on Apache Axiom) and the Apache Tomcat Server (Apache, 2008).

- An XML client API including WSDL and policy support
- Support for various message exchange patterns
- Synchronous and asynchronous function calls
- WS-Policy driven code generation
- A flexible service lifecycle model
- Support for SOAP, WSDL, WS-Reliable Messaging, WS-Security, WS-Addressing and SAAJ.

In addition, Axis2 provides data bindings that enable application developers to generate SOAP messages without concern for constructing or parsing them.

As pointed out previously, although our implementation is specific to the purchase and manufacturing of a customized computer, the underlying MIDAS system is general and can be used for other kinds of products.

3.1 The customer side

Customers can access the MIDAS system and the manufacturer's catalog of products using their Web browsers. The customers are represented in the manufacturer's database, and authentication is required before the resources for a particular customer can be accessed.

At the manufacturer, servlets use the Customer Web Service interface and make Web Service calls to MIDAS through the API provided. Figure 5 shows an example SOAP reply message for a catalog request.

```
POST /axis2/services/MidasWS HTTP/1.1
Content-Type: application/soap+xml; charset=UTF-8; action="urn:receiveOrder"
User-Agent: Axis2
Host: vanguard.ece.ucsb.edu:9904
Transfer-Encoding: chunked

<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
 <soapenv:Body>
  <ns2:receiveOrder xmlns:ns2="http://ws.midas.scc06">
   <ns2:param0>
    <ns1:clientID xmlns:ns1="http://data.midas.scc06/xsd">13783459</ns1:clientID>
    <supplies xmlns="http://data.midas.scc06/xsd">
     <amount>1</amount>
     <brand>Intel</brand>
     <id>2</id>
     <name>Intel Core 2 Duo T5550 (2MB cache/1.83GHz/667Mhz FSB)</name>
     <price>110.0</price>
    </supplies>
    <supplies xmlns="http://data.midas.scc06/xsd">
     <amount>1</amount>
     <brand>Brand</brand>
     <id>8</id>
     <name>4GB Shared Dual Channel DDR2 SDRAM at 667MHz</name>
     <price>600.0</price>
    </supplies>
    <supplies xmlns="http://data.midas.scc06/xsd">
     <amount>1</amount>
     <brand>Brand</brand>
     <id>12</id>
     <name>Item name 1</name>
     <price>200.0</price>
    </supplies>
```

Fig. 5. Example Web Service call to retrieve product catalog information

Once the customer is provided with the product catalog, the customer can create his own customized computer by selecting a particular supply for each component. Figure 6 shows an example SOAP request message for the customer's purchase order to MIDAS.

3.2 The manufacturer side

3.2.1 The Orders Database

The Orders Manager and the Materials Manager at the manufacturer use the Orders Database to insert, query and update the status of customer orders. The Orders Database includes the following tables:

- **Supply Table:** Represents alternative supplies for components
- **Component Table:** Represents the components of materials
- **Materials Table:** Keeps information about the materials that the manufacturer offers to customers
- **SupplyComponent Table:** Relates supplies and components
- **ComponentMaterial Table:** Relates components and materials

```
HTTP/1.1 200 OK
Date: Tue, 01 Apr 2008 02:30:23 GMT
Server: Simple-Server/1.1
Transfer-Encoding: chunked
Content-Type: application/soap+xml; charset=UTF-8; action="urn:getMaterialsResponse"

<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
 <soapenv:Body>
  <ns:getMaterialsResponse xmlns:ns="http://ws.midas.scc06"
    xmlns:ax21="http://data.midas.scc06/xsd">
   <ns:return type="scc06.midas.data.Material">
    <ax21:items type="scc06.midas.data.Component">
     <ax21:items type="scc06.midas.data.Supply">
      <ax21:brand>Intel</ax21:brand>
      <ax21:id>1</ax21:id>
      <ax21:name>Intel Core 2 Duo T5450 (2MB cache/1.66GHz/667Mhz FSB)</ax21:name>
      <ax21:price>100.0</ax21:price>
     </ax21:items>
     <ax21:items type="scc06.midas.data.Supply">
      <ax21:brand>Intel</ax21:brand>
      <ax21:id>2</ax21:id>
      <ax21:name>Intel Core 2 Duo T5550 (2MB cache/1.83GHz/667Mhz FSB)</ax21:name>
      <ax21:price>110.0</ax21:price>
     </ax21:items>
     <ax21:items type="scc06.midas.data.Supply">
      <ax21:brand>Intel</ax21:brand>
      <ax21:id>3</ax21:id>
      <ax21:name>Intel Core 2 Duo Processor T7250 (2.0GHz/800Mhz FSB, 2MB</ax21:name>
      <ax21:price>150.0</ax21:price>
     </ax21:items>
```

Fig. 6. Example Web Service call to purchase a customized computer

- **Quotes Request Table:** Keeps information about the Quote requests sent
- **Quotes Reply Table:** Keeps information about the Quote replies received
- **Sales Table:** Keeps information about the customers' order information. For each order item, new sales information is created.

Figure 7 shows the relationships between the tables for supplies, components and materials.

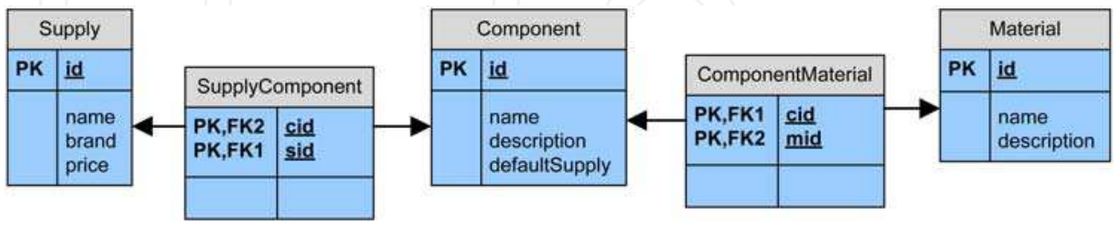


Fig. 7. The relationship between the database tables of MIDAS

3.2.2 The Registry and Registry Database

For the implementation of the UDDI Registry, we have used jUDDI (JUDDI, 2008), which runs with Jakarta Tomcat. The Registry is where the suppliers register their information (using the API provided) and where the manufacturers find that information (using the API provided). The suppliers update their information using authentication provided by jUDDI.

The UDDI Registry is backed with a MySql Server. The Registry accesses the MySql Database using JDBC. The UDDI Registry is deployed on one of the PlanetLab Nodes. The use of PlanetLab is discussed in more detail below.

4. Performance evaluation

In Section 2.2.2 we discussed different strategies that the DB Manager can use in the Waiting phase to aggregate orders from customers before triggering the Quotes Manager to communicate with suppliers. In Section 2.2.3 we discussed different strategies that the Quotes Manager can use in the Quotes phase to accumulate Quote replies from 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 used the PlanetLab global research network (PlanetLab, 2008). Figure 8 shows the deployment of MIDAS on our local network, with suppliers waiting for orders and customers sending their purchase orders from PlanetLab nodes across the United States.

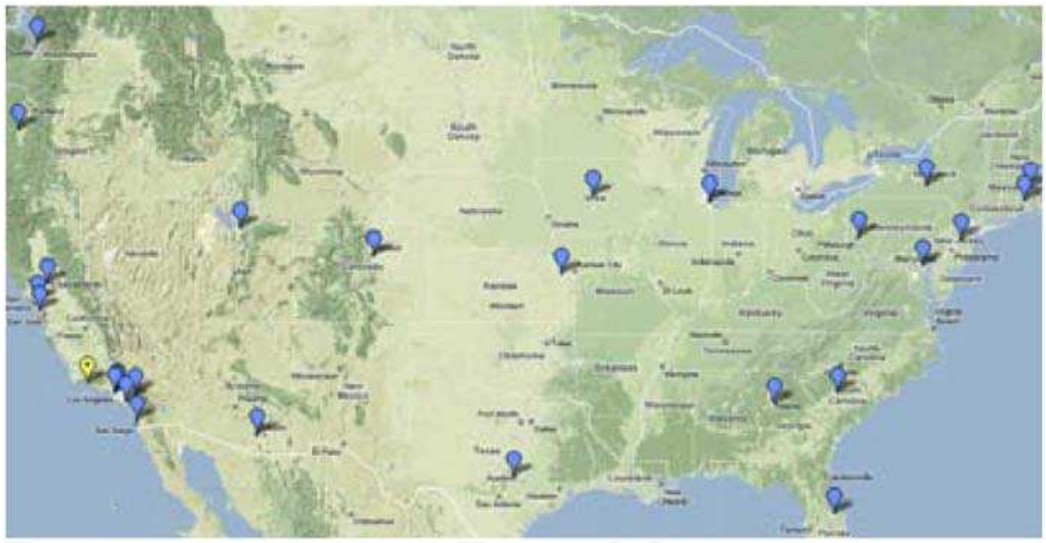


Fig. 8. Deployment of MIDAS on our network and on PlanetLab nodes across the USA

We evaluated the time spent in both the Waiting phase and the Quotes phase of MIDAS, *i.e.*, the time to aggregate orders and the time to aggregate Quote replies, under the different strategies. In addition, we investigated the probability that MIDAS makes a decision for the best quote possible. Finally, we calculated the average time needed to complete a customer order under the different strategies.

4.1 Order process

Before presenting the performance evaluation results, we present customer order timelines for a customized computer in order that the evaluation can be better understood.
Figure 9 depicts the timeline of a customer order submitted to MIDAS in the Waiting phase. The customer retrieves the product catalog for a computer from the manufacturer. Using this catalog, the customer creates the customized computer and submits his order to MIDAS. In this case, the order includes five different supply items (processor, graphics card,

memory, hard disk, optical disk drive) for the components of the customized computer. On receiving orders from the customers, MIDAS first aggregates the supply items ordered, and then contacts the suppliers when the total number of items for a particular supply reaches a threshold or, alternatively, when a timeout occurs. Depending on the strategy used, the aggregation time for a supply varies. Frequently chosen supplies are aggregated faster than infrequently chosen supplies. The performance evaluation for the Waiting phase is discussed in Section 4.2.

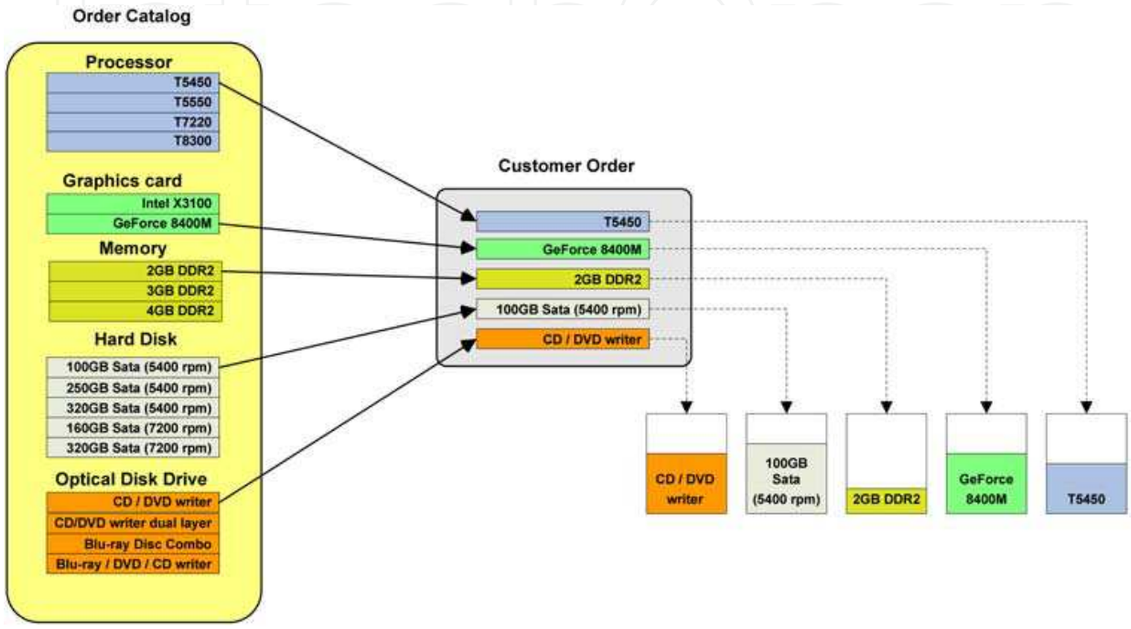


Fig. 9. Waiting Phase. Timeline of a customer order

Figure 10 depicts the timeline of the customer order for the Quotes phase. After deciding to contact the suppliers for a specific supply, MIDAS retrieves information about the respective suppliers for that supply from the UDDI Registry. MIDAS submits a Quote request for that supply to those suppliers. A supplier makes a callback to deliver its Quote reply. Once a supplier has finished the process of making a reservation related to the Quote request, it sends a Quote reply using the callback Web Service. MIDAS collects these Quote replies and makes a decision about the status of a quote, which affects the status of different supply items. The selected supplier is notified with a confirmation related to the purchase. The suppliers not chosen are contacted with a cancellation of the reservation. Once a decision for all order items for a customer order is made, the customer order is complete. The performance evaluation for the Quotes phase is discussed in Section 4.3.

4.2 The Waiting phase

As discussed in Section 2.2.2, the Waiting phase of MIDAS is based on one of several strategies. Here we consider:

- Aggregating a threshold number of orders
- Timeout for a specific amount of time
- Hybrid of threshold number of orders and timeout.

We evaluated these strategies under different customer order arrival rates. In our evaluation, the manufacturer receives a total of 1000 customer orders.

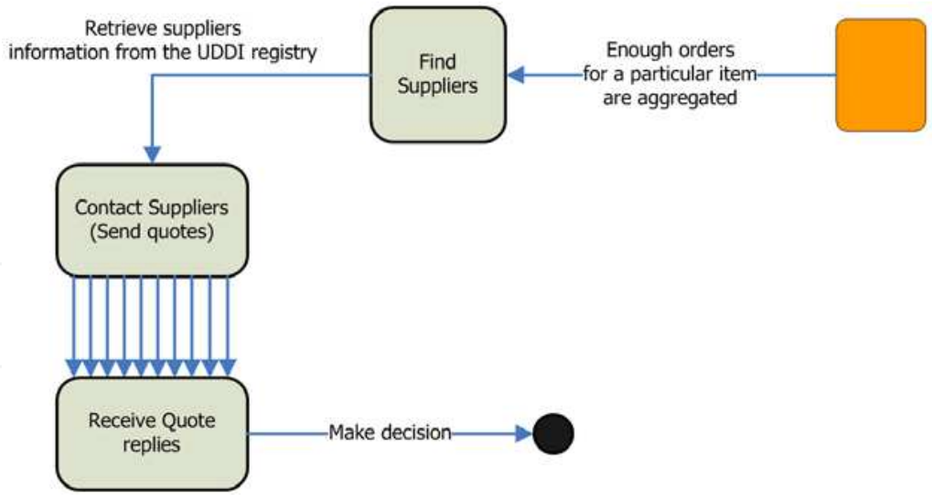


Fig. 10. Quotes Phase: Timeline of a customer order

4.2.1 Aggregating a threshold number of orders

This strategy aggregates orders for a specific number of items for a particular supply before the Quotes Manager issues Quote requests to the suppliers.

We evaluated the time spent to aggregate orders for 5, 10, 15, and 20 supply items with different customer order arrival rates (customer order arrives every 4, 6, 8, 10, and 12 seconds) and for the same number of alternatives (four alternatives) for each of the different components from which the customer can choose. The results are shown in Figure 11.

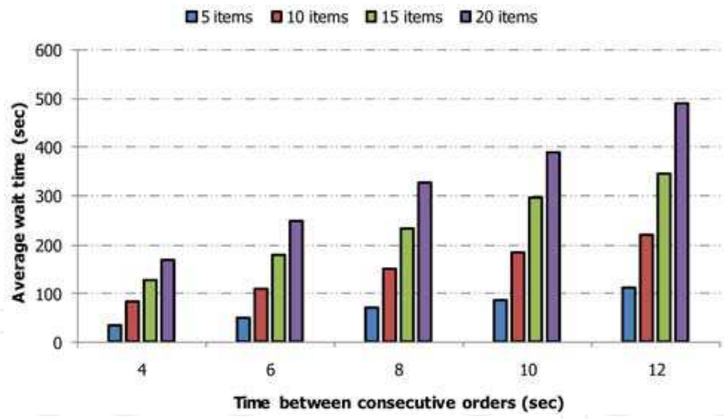


Fig. 11. Order aggregation with different customer order arrival rates and four alternatives for each component

As Figure 11 shows, the average wait time to aggregate enough supply items is higher for low customer order arrival rates. Moreover, the increased threshold delays the time to complete the Waiting phase. The threshold number of orders aggregation strategy keeps the number of supply items at a certain limit; however, a low customer order arrival rate might delay the order of a particular supply if that supply is not popular.

For variable numbers of alternatives for the different components (*e.g.,* 2 choices for the processor, …, 6 choices for the hard disk), the weighted average of the wait time during order aggregation is essentially the same as that for the same number of alternatives. This

behavior is expected because a supply that is aggregated faster has a larger number of occurrences but a smaller wait time.

Customers will be satisfied if their orders are completed earlier. However, contacting suppliers with orders for more supply items is more favorable for the manufacturer than contacting the suppliers with orders for fewer of them. The more supply items purchased from the supplier, the more gain the manufacturer has. Thus, the manufacturer faces a conflict as to whether to be concerned about the number of supply items to aggregate (manufacturer gain) or to limit the wait time for aggregation of supply items (customer satisfaction).

4.2.2 Timeout for a specific amount of time

The timeout strategy keeps the wait time for orders at a certain level. Therefore, we consider the number of orders aggregated as a function of the customer order arrival rate and the timeout value. The results for the timeout strategy are presented in Figure 12.

As Figure 12 shows, an increased timeout value results in the aggregation of a larger number of items for a particular supply. For the same timeout value, the customer order arrival rate affects the number of orders aggregated before the timeout occurs.

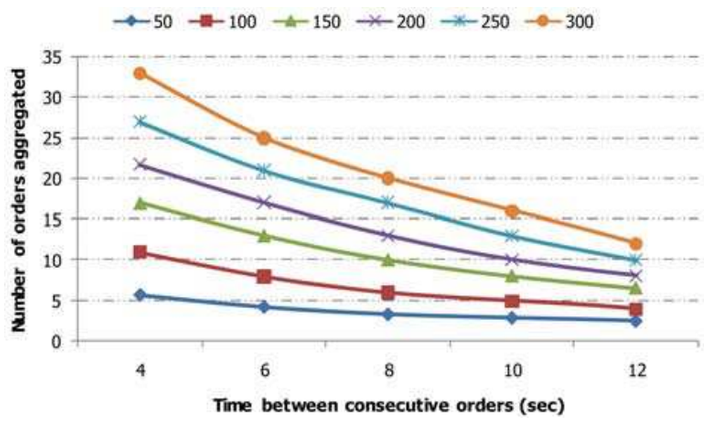


Fig. 12. Order aggregation with different timeout values

4.2.3 Hybrid of threshold number of orders and timeout

Although we have two strategies for the Waiting phase, *i.e.*, aggregating a threshold number of orders and waiting for a timeout to occur , it is better to use both particularly if the customer's request rate is low or is not known a priori. Taking both strategies into account can obtain the threshold number of orders in less than the maximum wait time of the Waiting phase but still bound the wait time to allow the Quotes phase to proceed.

4.3 The Quotes phase

As discussed in Section 2.2.3, the Quotes phase of MIDAS is based on one of several strategies. Here we consider:

- Aggregating a threshold percentage of Quote replies
- Timeout for a specific amount of time
- Hybrid of threshold percentage of Quote replies and timeout.

In the Quotes phase, MIDAS contacts the relevant suppliers and submits a Quote request to each of them. On receiving a Quote request, a supplier incurs some processing time before it replies to MIDAS using the callback Web Service endpoint passed in the Quote request. The manufacturer collects Quote replies concurrently in an asynchronous manner, and the suppliers respond to Quote requests after a processing time that depends on the particular type of supplier.
The processing time for the suppliers, shown in Figure 13, is given by the following formula:

$$Processing\ time \ = \ 20 - 5 * \frac{\log\ (random)}{\log\ (factor)}$$

The processing time depends on a random number between 0 and 1, and corresponds to processing times between 20 seconds and ∞. With a high probability, the processing time is close to 20 seconds, and with a very low probability, the processing time is ∞, which represents a supplier that crashes after receiving a Quote request from MIDAS.

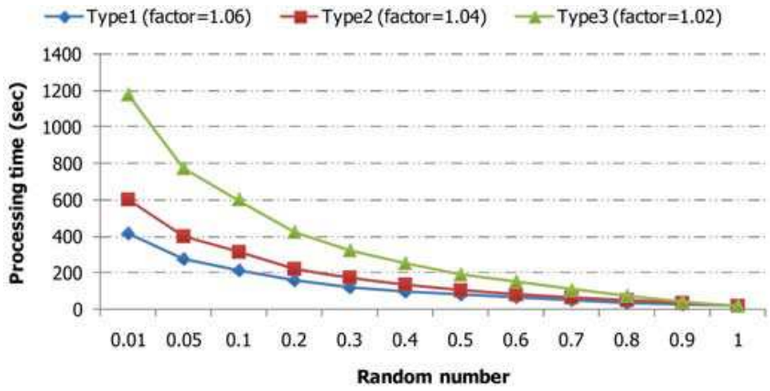


Fig. 13. Modeling of the processing time for the Suppliers

The processing time also depends on a scaling factor to represent different types of suppliers with different processing times. The factors 1.06, 1.04, and 1.02 correspond to processing times in the ranges 20-400, 20-600, and 20-1200. Type 1 suppliers (factor=1.06) respond to Quote requests earlier than Type 3 suppliers (factor=1.02), and Type 2 suppliers respond to Quote requests in between the other two.
The time spent before determination of the status of a quote depends on the strategy and the type of supplier (1, 2, or 3). Depending on the strategy, it is possible to decide the status of a quote before receiving all of the Quote replies. Aggregation of more Quote replies increases the chance of selecting the best quote.
For each of the above strategies in the Quotes phase, we used the threshold strategy in the Waiting phase to evaluate the average wait time and the best quote ratio. The best quote ratio is the probability that MIDAS decides the best possible Quote reply.

4.3.1 Aggregating a threshold percentage of Quote replies

In this strategy, if the number of Quote replies exceeds a certain percentage, the Quotes Manager stops collecting Quote replies from the suppliers and completes the customer orders. Compared to the strategy of collecting all Quote replies, this strategy decreases the customer's response time (increases the customer's satisfaction), at the cost of reducing the best price ratio.

Figure 14 shows the average wait time the threshold percentage strategy in the Quotes phase. Depending on the number of suppliers that the Quotes Manager contacted, the Quotes Manager waits for the number of Quote replies given by:

$$Expected\ replies = Requests\ sent * Percentage\ to\ aggregate$$

The results show a linear increase and then an exponential increase for the case that MIDAS waits to collect all Quote replies. The average wait time is dominated by the maximum processing time from one of the suppliers for that quote. The average wait time can be kept reasonable, although MIDAS must wait for 90% of the expected Quote replies.

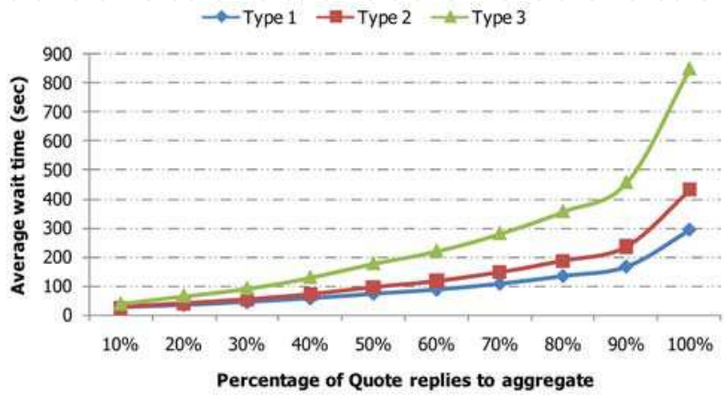


Fig. 14. Average wait time for the threshold percentage strategy

We also evaluated the best quote ratio against the percentage of expected Quote replies for the threshold percentage strategy. The results indicate that the best quote ratio is independent of the processing time at the suppliers, *i.e.*, the type of supplier.

4.3.2 Timeout for a specific amount of time

When the Quotes Manager collects Quote replies from the suppliers, it waits for a specific amount of time, $delay_{AQ}$. When the timeout occurs, the Quotes Manager decides on a supplier for a specific supply for the customer's order and ignores late Quote replies. Because the Quotes Manager doesn't need to collect Quote replies from all of the suppliers, this strategy decreases the customer's response time (increases the customer's satisfaction), compared to the strategy of collecting all Quote replies. On the other hand, the Quotes Manager cannot always obtain the best price from the suppliers, because the Quotes Manager might receive the Quote reply with the best price too late and thus ignore it.

The timeout strategy decides the status of a quote, either when a timeout occurs, or before the timeout when all of the expected Quote replies are received. As expected, our evaluation of the average wait time, *i.e.*, $delay_{AQ}$, shows that the average wait time increases as the time spent aggregating Quote replies increases. Moreover, the average wait time stabilizes around the maximum processing time of the suppliers. After stabilization, increasing the timeout does not affect the time taken to make a decision for the quote.

More interesting is our evaluation of the best quote ratio for the timeout strategy, *i.e.*, the probability that MIDAS catches the best quote given a particular timeout value, as shown in Figure 15. When the Quotes Manager decides the status of a quote early, the probability of catching the best quote is low because a better Quote reply might arrive later. Type 1 suppliers that have a small processing time send their Quote replies early, which increases

the number of Quote replies aggregated in the Quotes phase. Once the best quote ratio reaches a maximum value, increasing the timeout does not affect the best quote ratio.

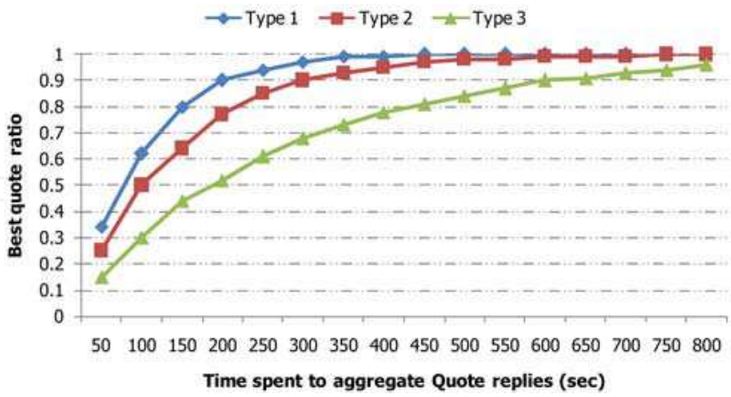


Fig. 15. Best quote ratio for the timeout strategy

4.3.3 Hybrid of threshold percentage of quote replies and timeout

Although we have two strategies for the Quotes phase, *i.e.*, aggregating a threshold percentage of Quote replies and waiting for a timeout to occur, it is better to use both strategies, particularly if the supplier's response rate is low. Taking both strategies into account can obtain the threshold percentage of Quote replies in less than the maximum wait time of the Quotes phase but still bound the wait time of the Quotes phase of MIDAS.

4.4 Customer order analysis

Finally, we provide a customer order analysis in terms of the wait time that elapses before completion of a customer order, *i.e.*, all of the supply items needed for the customer order have been ordered. Thus, the order completion time is the maximum of the completion times for all supply items i needed to fill the customer order, which is given by:

$$Order\ completion\ time \quad = \quad max_{\forall i}\ (Delay_{AO}(i) + Delay_{AQ}(i))$$

The results of the analysis for the timeout strategy are shown in Figure 16 for the case in which MIDAS receives orders every 10 seconds and 5 items are aggregated before MIDAS enters the Quotes phase.

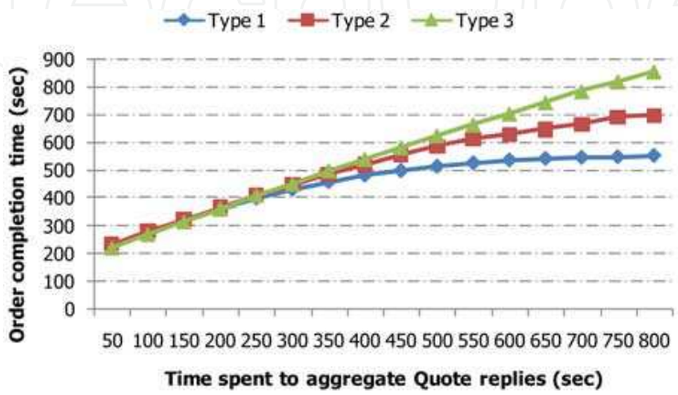


Fig. 16. Order completion time for the timeout strategy

Under the same conditions, we investigated the customer order completion time using the threshold percentage strategy in the Quotes phase. The order completion time looks similar to the behavior seen for the wait time using the threshold percentage strategy in the Quotes phase shown in Figure 14.

From these experimental results, we draw the following conclusions. To enhance the satisfaction of the customers 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. Thus, a conflict exists between the customer and the manufacturer, so a balance point must be found. The results also show that the Waiting phase could be adjusted to provide the minimum response time by using a hybrid scheme based on both the threshold and timeout strategies. The analysis of the Quotes phase shows that the manufacturer does not really need to aggregate all possible Quote replies. Above the best quote ratio, increasing the threshold percentage of Quote replies aggregated or the timeout does not affect the manufacturer's gain considerably, but delays the customer's order completion time. The strategies used in the Quotes 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, again, not all of the Quote replies are taken into account. In the Quotes phase, both the threshold and timeout strategies reduce the customer response time at the cost of decreasing the best price ratio. Again, a hybrid strategy is preferable.

5. Related work

The MIDAS system is an application of the *enterprise-on-demand* model (Stone, 2004), which aims to bridge the gap between business and technology utilizing Business Process Management and the Service Oriented Architecture. Some predict that the enterprise-on-demand model will have a greater impact than the client-server model and will be the killer enterprise application for the Internet.

We have focused in this paper on information flow in supply chains, as realized by MIDAS. Wu *et al. (*2005) have also considered information flow in supply chains. They observe that, in a competitive e-business environment, an enterprise must be involved in managing the supply chain from both the upstream side (the suppliers) and the downstream side (the customers). They propose a novel approach that brings the business processes and services together to support supply chain management.

As noted previously, MIDAS aims to automate the supply chain and, thus, to reduce human intervention, errors and costs, resulting in a more efficient supply chain. Dong and O'Brien (1999) have a similar objective, but they base their business model on the four criteria: 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, they set objectives associated with the criteria for each supply chain stage in order to satisfy customer service targets and to select the best supply chain management strategy. At the operation level, they optimize manufacturing and logistics activities for the given targets.

Jinho and Rogers (2005) 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 describe a system that demonstrates their approach, as we do for MIDAS.

Yang *et al.* (2005) have investigated customization and postponement in supply chains to reduce costs, realize diversity, and improve agility. They consider four kinds of postponement in supply chains: supply, manufacture, delivery, and service. Waller *et al.* (2000) also investigated supply chain customization, and discuss market orientation with supply chain customization. Zhou *et al.* (2003) have also investigated customization in supply chains. They address *mass customization*, which supports customer innovation and which integrates mass production and customized production.

MIDAS realizes the concept of on-demand, just-in-time manufacturing to reduce inventory carrying costs and business logistics costs, by maintaining logical inventory, rather than physical inventory in warehouses. Tanik *et al.* (2001) have proposed a zero-time framework based on what they call the T-strategy, which allows enterprises to adapt in a timely manner to changing market conditions.

In our performance evaluation of MIDAS, we have considered the customer's satisfaction (measured by the average customer response time) and the manufacturer's gain (measured by the number of orders aggregated or the best price ratio). In contrast, Li *et al.* (2005) have considered the average supplier response time in their evaluation of supply chains. Giglio and Minciardi (2003) have investigated the modeling and optimization of supply chains that involve multiple production sites and multiple suppliers, using mathematical programming techniques that aim to minimize costs in the network. Zhao and Jin (2005) have investigated optimized coordination of supply chains based on relationships and dependencies.

Wang *et al.* (2004) observe that appropriate supply chain partners have a large effect on the output value of a supply chain system. They propose an Internet-driven electronic marketplace that can provide an effective platform to select the right partners. In some ways, such a marketplace is like the MIDAS Registry. They present a model of procurement strategies, and discuss factors for the success of such a marketplace.

There exist other papers on build-to-order supply chain management that focus on business and conceptual issues, rather than on the design and implementation of an automated supply chain management system. In particular, Gunasekarana and Ngai (2005) present a review of build-to-order supply chain management and a framework for development. They note that Dell, Compaq and BMW are the best examples of companies that have applied this strategy successfully. Graham and Hardaker (2000) highlight the role of the Internet in building flexible, agile, on-demand supply chains based on the build-to-order strategy. They note that the build-to-order strategy not only addresses diverse customer needs, but also lowers inventory stocks so that parts are pulled from the suppliers as needed, with almost no in-process inventory.

Ghiassi and Spera (2003) discuss industry solutions and best practices for Internet-based supply chains that support mass customization. They emphasize the success of Dell in switching to the build-to-order strategy and gaining a 160% return on its investment. Lancioni *et al.* (2000) discuss the use of the Internet for managing the major aspects of supply chains, particularly for order processing, purchasing/procurement and transportation. They note that enterprises use the build-to-order strategy to achieve lower inventory costs (with somewhat higher production costs), track inventory more accurately, and report the status of orders. Fontanella (2000) observes that Internet technologies make foreign markets more accessible and make it easier to integrate foreign customers, suppliers and intermediate companies into the supply chain, increasing savings and providing innovation.

Forza and Salvador (2002) discuss the difficulties in managing build-to-order systems. They focus particularly on the challenges due to custom customer orders, which must be

maintained and processed carefully. Kolish (2000) also discusses the difficulties of the build-to-order model with the production of ships, airplanes and large-scale machine tools. In particular, he focuses on the coordination of fabrication and assembly with respect to scarce capacities.

6. Conclusions and future work

In this article we have presented the design and implementation of the MIDAS system for automated supply chain management. MIDAS allows customers, manufacturers, and suppliers to cooperate over the Internet and the World Wide Web. It aims to meet the needs of the customers on time, and to reduce the costs of the manufacturer by eliminating the need for a large inventory. As a Service Oriented Architecture based on Web Services, MIDAS increases business flexibility and enables businesses to adapt more quickly to changing business needs.

At the manufacturer, MIDAS uses two phases in processing orders from the customers: The Waiting phase in which it aggregates orders from the customers before it makes Quote requests from the suppliers, and the Quotes phase in which it collects Quote replies from the suppliers before deciding on the supplier that will provide the particular supply. The manufacturer can use one of several strategies to decide how long to wait to collect orders from customers or quotes from suppliers. We have presented an evaluation of these strategies based on our implementation.

We have described the services of MIDAS related to information flows. Relationships and dependencies between the components that constitute a material 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. In the future, we plan to augment the MIDAS system with a workflow component that handles the relationships and dependencies between the components of a material. We also plan to design and implement the services related to product flows and financial flows.

The MIDAS system, 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. MIDAS might 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 rates at which the customers place orders, the DB Monitor can adjust its timeout or threshold values accordingly. The Quotes Manager can select better suppliers dynamically by waiting until it receives Quote replies from the suppliers whose Quote replies were previously ignored because they were too late.

7. Acknowledgements

This research was supported in part by the UC Discovery Grant Program and QAD, Inc., grant number COM05-10194.

8. References

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

Apache Foundation. Apache Web Server. http://httpd.apache.org/

Apache Foundation. Java implementation of the Universal Description, Discovery, and Integration (UDDI) specification for Web Services. 2008, http://ws.apache.org/juddi/

M. Champion, C. Ferris, E. Newcomer, and D. Orchard. Web Services Architecture. 2002. http://www.w3c.org/TR/2002/WD-ws-arch-20021114/

S. Chatterjee and J. Webber. *Developing enterprise Web Services: An Architect's Guide*. Prentice Hall, Englewood Cliffs, NJ, 2003

L. Dong and C. O'Brien. Integrated decision modeling of supply chain efficiency. *International Journal of Production Economics*, 59(1-3):147-157, March 1999

T. Erl. *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*. Prentice Hall, Englewood Cliffs, NJ, 2005

J. Fontanella. The Web-based supply chain. *Supply Chain Management Review,* 3(4):17-20, 2000

C. Forza and F. Salvador. Managing for variety in the order acquisition and fulfillment process: The contribution of product configuration systems. *International Journal of Production Economics,* 76(1):87-98, 2002

M. Ghiassi and C. Spera. Defining the Internet-based supply chain system for mass customized markets. *Computers and Industrial Engineering,* 45(1):17-41, 2003

D. Giglio and R. Minciardi. Modeling and optimization of multi-site production systems in supply chain networks. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 3, pages 2678-2683, October 2003

G. Graham and G. Hardaker. Supply-chain management across the Internet. *International Journal of Physical Distribution and Logistics Management,* 30(3/4):286-295, 2000

A. Gunasekarana and E. W. T. Ngai. Build-to-order supply chain management: A literature review and framework for development. *Journal of Operations Management,* 23(5):423-451, 2005

K. Jinho and K. Rogers. An object-oriented approach for building a flexible supply chain model. *International Journal of Physical Distribution and Logistics Management*, 35(7):481-502, July 2005

R. Kolisch. Integration of assembly and fabrication for make-to-order production. *International Journal of Production Economics,* 68(3):287-306, 2000

R. A. Lancioni, M. F. Smith, and T. A. Oliva. The role of the Internet in supply chain management: Logistics catches up with strategy. *Industrial Marketing Management,* 29(1):45-56, 2000

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*, 10(1):89-93, 2005

L. E. Moser and P. M. Melliar-Smith. Service Oriented Architecture and Web Services. *Encyclopedia of Computer Science and Engineering*, John Wiley & Sons, New York, NY, 2008

R. Murch. *Autonomic Computing.* On Demand Series, IBM Press, April 2004

E. Newcomer and G. Lomow. *Understanding SOA with Web Services.* Independent Technology Guides, Addison-Wesley Professional Series, December 2004

OASIS. Reference Model for Service Oriented Architecture, 2006. http://www.oasis-open.org/committees/tc home.php?wg abbrev=soa-rm

PlanetLab Global Research Network. http://www.planet-lab.org

A. Stone. Demanding Internet enterprise. *IEEE Internet Computing*, 8(3):13-14, 2004

U. Tanik, M. M. Tanik, and L. Jololian. Internet enterprise engineering. A "zero-time" framework based on "T-strategy". *Proceedings of the IEEE Southeast Conference*, pages 263-270, Clemson, SC, March 2001

M. A. Waller, P. A. Dabholkar, and J. J. Gentry. Postponement, product customization, and market-oriented supply chain management. *Journal of Business Logistics*, 2000

L. P.Wang, F. Y. Qiu, H. R. Dai, and Z. C. Chen. Relationships of electronic marketplace and partner selection of supply chain. *Computer Integrated Manufacturing Systems*, 10(5):550-555, 2004

B.Wu, M. Dewan, L. Li, and Y. Yang. Supply chain protocolling. *Proceedings of the 7th IEEE International Conference on E-Commerce Technology*, pages 314-321, July 2005

J. Yang, S. Zhao, and J. Wang. Study on postponement in customization supply chain. *Industrial Engineering and Management*, 10(4):35-44, 2005

T. Z. Zhao and Y. H. Jin. Optimized coordination of supply chains based on dependency. *Computer Integrated Manufacturing Systems*, 10(8):929-933, 2005

W. Zhao, F. Kart, L. E. Moser, and P. M. Melliar-Smith. A reservation-based extended transaction protocol for coordination of Web Services. *Journal of Web Services Research*, 5(3):64-95, 2008

D. Zhou, B. H. Xiang, and G. S. Zou. The strategy of mass customization and the countermeasure of Chinese enterprises. *Industrial Engineering and Management*, 8(5):12-16, 2003

IntechOpen



Supply Chain the Way to Flat Organisation

Edited by Julio Ponce and Adem Karahoca

ISBN 978-953-7619-35-0
Hard cover, 436 pages
Publisher InTech
Published online 01, January, 2009
Published in print edition January, 2009

With the ever-increasing levels of volatility in demand and more and more turbulent market conditions, there is a growing acceptance that individual businesses can no longer compete as stand-alone entities but rather as supply chains. Supply chain management (SCM) has been both an emergent field of practice and an academic domain to help firms satisfy customer needs more responsively with improved quality, reduction cost and higher flexibility. This book discusses some of the latest development and findings addressing a number of key areas of aspect of supply chain management, including the application and development ICT and the RFID technique in SCM, SCM modeling and control, and number of emerging trends and issues.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Firat Kart, Louise E. Moser and P. M. Melliar-Smith (2009). Managing and Integrating Demand and Supply Using Web Services and the Service Oriented Architecture, Supply Chain the Way to Flat Organisation, Julio Ponce and Adem Karahoca (Ed.), ISBN: 978-953-7619-35-0, InTech, Available from: http://www.intechopen.com/books/supply_chain_the_way_to_flat_organisation/managing_and_integrating_demand_and_supply_using_web_services_and_the_service_oriented_architecture

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen