WEB SERVICES ENABLED PROCUREMENT IN THE EXTENDED ENTERPRISE: AN ARCHITECTURAL DESIGN AND IMPLEMENTATION

Minder Chen
School of Management, George Mason University, Fairfax, VA
mchen@gmu.edu

Mary J. Meixell
School of Management, George Mason University, Fairfax, VA
mmeixell@gmu.edu

ABSTRACT

Earlier supply chain implementations in EDI and XML have not fully supported the real time and dynamic requirements of e-procurement, in large part due to the limitations of the technologies available. The purpose of this paper is to illustrate how Web services can be used to implement a real time and dynamic supply chain operation. In particular, we focus on an e-procurement application in the context of business-to-business (B2B) supply chain integration. A prototype system has been developed to illustrate how Web services can enhance performance of the firms engaged in B2B procurement. The architecture and design of the system presented in this paper are based on a service-oriented architecture and on a three-tier enterprise design pattern. The impacts and implications of using Web services to enable real-time and dynamic business processes are also discussed.

Keywords : Web services, e-procurement, dynamic e-business, supply chain management

1 Introduction

Web services provide a standard-based approach to implementing distributed components. They offer data and business logic services over standard protocols such as HTTP, XML, and SOAP (W3C, 2000a) over the Internet. The Stencil Group, an IT Research firm, proposed a broader definition of Web services (Stencil Group, 2001):

"Loosely coupled, reusable software components that semantically encapsulate discrete functionality and are distributed and programmatically accessible over standard Internet protocols."

The term "loosely coupled" implies that Web services are independent of any programming languages, platforms, and object models. Using the ubiquitous and low-cost Internet, Web services can easily provide interoperable software functions over the Intranet and the Internet. From a distributed computing architecture viewpoint, Web services describe a service-oriented and component-based application architecture. From a business-oriented and conceptual viewpoint, Web services are an enabling technology that may be used to build a dynamic e-business model. The concomitant e-business processes are composed of discrete tasks implemented as Web services, and distributed widely throughout a network of value-added services.

A number of companies in a variety of industries have successfully integrated their supply chains with noteworthy performance improvements. These early adopters are in consumer goods (Camm, Chorman, Dill, Evans, Sweeney, and Wegryn, 1997), automotive (Hahn, Duplaga, and Hartley, 2000) and electronics (Lin, Ettl, Buckley, Bagchi, Yao, Naccarato, Allan, Kim, and Koenig, 2000) industries. By uniting their disparate business processes, the trading partners in these supply chains have experienced performance improvements in transaction and production related costs, in asset utilization, but also importantly in responsiveness to customers needs. Therefore, companies are motivated to explore innovative and integrated applications within as well as among enterprises in the supply chain. Deploying Web services reduces the integration costs associated with achieving an extended enterprise, as well as improves service levels that would otherwise not be achievable (Adams, 2003; Samtani and Sadhwani, 2002). In the real time dynamic enterprise, information from internal processes or from external trading partners (both customers and suppliers) needs to become available faster to support changes in the environment. In the research presented in this paper, we specifically address the supplier-facing procurement process, and in particular the disruptions that can and often do occur in the daily operation of a real world supply chain.

In the next section, we discuss the evolution of technology in the context of the supply chain by assessing the differences between EDI, XML, and Web services. Section 3 discusses the architecture and design of a Web service enabled e-Procurement application in the supply chain, and provides a description of an e-procurement process that utilizes the unique dynamic binding features of Web services. Following that is a discussion of a prototype that implements Web services to support dynamic e-Procurement. In Section 5 we analyze the operational impacts of Web services in this procurement context, followed in section 6 by a discussion of challenges and future research directions.

2 Symbiotic Relationship and Evolution between Supply Chain Management and Information Technology

Supply Chain Management (SCM) is an approach to coordinating the functions and processes among enterprises that are involved in order fulfillment, with the objective of delivering what the final customer wants at the time and place the customer desires it, in a manner that minimizes total costs for the producing organizations. A supply chain typically extends across multiple enterprises including suppliers, manufacturers, transportation carriers, warehouses, retailers as well as the customers themselves (Chopra and Meindl, 2004). From a process viewpoint, SCM coordinates order management, production management, inventory management, purchasing, distribution, transportation, and product design. In the context of purchasing or procurement, extending processes across enterprise lines involves collaboration such as early supplier involvement in product design, as well as ongoing daily coordination of purchased material flows. SCM in this last context typically entails sharing forecast, order, inventory, and production information to better coordinate management decisions at multiple decision points throughout the extended enterprise.

From an information technology perspective, SCM synchronizes a set of interrelated activities across multiple organizational boundaries with different computing platforms and data formats, making the integration of these activities into a streamlined and efficient process a major challenge. Thus, information technologies and related standards have made major impacts on the evolution of supply chain management. The evolution from EDI-based static supply chains towards Web services-enabled real-time and dynamic supply chains is discussed in this section.

2.1 EDI

Electronic commerce as we know it before 1995 is synonymous with Electronic Data Interchange (EDI). EDI has been in use by major retailers, manufacturing firms and logistics companies, initially to reduce transaction costs with trading partners. In more recent years, some supply chains have integrated processes using EDI (Davis, 1995; Robins, 1995; Mitchell, 2001; Varon, 2001). Wide application of supply chain integration has been elusive using EDI, however, partly because of the timeliness problem with batch transmissions. Batch processing typically makes it difficult or impossible to have consistent data across supply chain locations in a timely manner.

This lack of synchronization is due to the additive nature of a set of organizations each managing their business processes in a highly dynamic fashion. In each firm within a supply chain, it is common for production, marketing, and engineering groups to continually update and improve their plans for the components and final products. Production planners may change the operating plan to improve efficiency, marketing frequently adjusts promotion and pricing to best respond to opportunities in the market, and engineering continually enhances the design of the product based on new information from the supplier and customer communities. In light of this, it is easy to see how no two locations in the chain will have the same set of complete data at any single point in time when data are transmitted and processed in batch mode.

EDI may be managed by the MIS department internal to the firms, but many companies have relied on value-added network (VAN) vendors to provide data transmission services. The EDI translators are needed within both the sending and the receiving companies to convert business documents or transactional data to EDI format and back to the data format usable by internal data systems. These VAN services are expensive so are the development costs of EDI translators. Because of this, the overall costs of implementing EDI are quite high. Accordingly, the adoption of EDI among small and medium enterprises is limited (Turban, 2002). Especially with high tech firms, the absence of these trading partners is a substantial loss to supply chain integration, as many emerging technologies have been developed within smaller entrepreneurial firms. The Internet and the Web have made the transmission of EDI-based documents more affordable (Zilbert, 2000). However, the building of EDI document parsers or translation software is still costly and error prone (Fu, 1999) because the EDI standards for encoding documents (i.e. ANSI X.12 and EDIFACT) are complicated due to their emphasis on conciseness.

2.2 XML

XML is a data interchange standard that emerged in the late 1990's to replace EDI standards in B2B e-commerce. XML should be considered as a suite of standards. XML, which stands for "eXtensible Markup Language", is considered extensible because it is not fixed format like EDI and HTML. XML has become the standard for portable data. There are three levels of abstraction related to XML (Chen, 2003): (1) XML definition languages which have two standards: XML 1.0 Recommendation (W3C, 2000b) and XML Schema (W3C, 2001a), (2) application domain standards - such as ones standardized by OASIS - for various business documents or data sets defined in XML DTD or XSD, and (3) the actual business documents that are encoded in XML format.

Additional supporting standards such as XSL (W3C, 2000c), XML DOM (W3C, 2000c), or SAX (SAX Project, 2002) are used for XML document parsing and translation.

There are several benefits of XML over EDI as follows:

1. Unicode: XML uses Unicode so that it can be encoded in different national languages.
2. Error checking: XML DTD and XSD files are basically definitions of markup languages. They define the structures and constraints of specific documents and therefore, can be used to capture and validate business documents, a critical function for B2B e-commerce.
3. Free and integrated parsing tools: Many free XML parsers are readily available in various programming languages and are fully integrated with enterprise computing platforms such as .NET and J2EE. Developers can easily use these XML tools to build applications to handle XML data.

Industries that have adopted XML in their supply chains include the retail industry and their consumer goods suppliers. In many firms, CPFR (Collaborative Planning, Forecasting, and Replenishment) integration initiatives have been built upon the XML platform (CPRF.ORG, 2002; Jouenne, 2000; Koloszyc, 1998). Accordingly, much of the gain achieved in supply chain integration to date is based on the enhancements provided by this technology over earlier communication technologies that include EDI, fax, and phone. However, EDI- or XML-based business documents are often processed in batch mode with processing delays such that up-to-date business data from trading partners may not be available to support real time decision making. So, neither EDI nor XML fully supports efforts to integrate supply chains in a real-time fashion.

2.3 Web Services

Web services typically use XML for wire data format and SOAP for messaging (i.e. XML Messaging Protocol) to support interoperable software components. Web services can be used to develop new software components or to build wrappers to expose business processes to external trading partners. It can also be used internally to provide programmable interfaces to legacy systems, and to integrate Web-enabled applications directly with some legacy systems (Samtani and Sadhwani, 2002a; Vecchio, 2001).

EDI and XML are both based on a data interchange approach to integrating supply chain processes, so that data is collected and sent to a trading partner to enable that organization to use it at will. Even with the collaborative forecasting function in CPFR, data is sent back and forth between customer and supplier, sometimes with markups of changes to enable the trading partner to update their own files with information held elsewhere in the supply chain. XML is still used for the data interchange purpose under SOAP, however, Web services take a service- and process-oriented approach in addressing integration issues in supply chain management. An application enabled with Web services can provide streamlined capabilities by consuming Web services from legacy systems (such as for checking inventory status) and external suppliers (such as requesting price quotes from suppliers). These Web services are composed transparently behind the scene in the application to give users access to more supply chain functions from a single application.

SOAP, a Web services messaging mechanism, follows the HTTP request/response message model. Therefore, operational data in the supply chain can be exchanged in real-time. Most Web services applications developed today use static binding to call specific Web services known at design time. Supply chain applications are however inherently dynamic, as conditions across the tiers change over time. For example, inventory levels of finished goods will continually fluctuate, as production increases and shipments decrease the stock level for finished goods. Using Web services, a supplier may request inventory status of a customer stocking location at the point in time an inventory replenishment decision is to be made. Web services, then, can be used to advance the operation of previous information technologies to enable real time data updates to dynamic supply chain conditions, providing necessary support that could not otherwise be provided efficiently to the extended enterprise. A dynamic binding approach is required in such a case; however, few literature sources discuss how to design and implement a Web services application with dynamic binding. We illustrate in the next section an architecture and a design for accomplishing this real time inter-connectedness in an extended enterprise.

3 An e-Procurement System via Dynamic Web Services: Architecture and Design

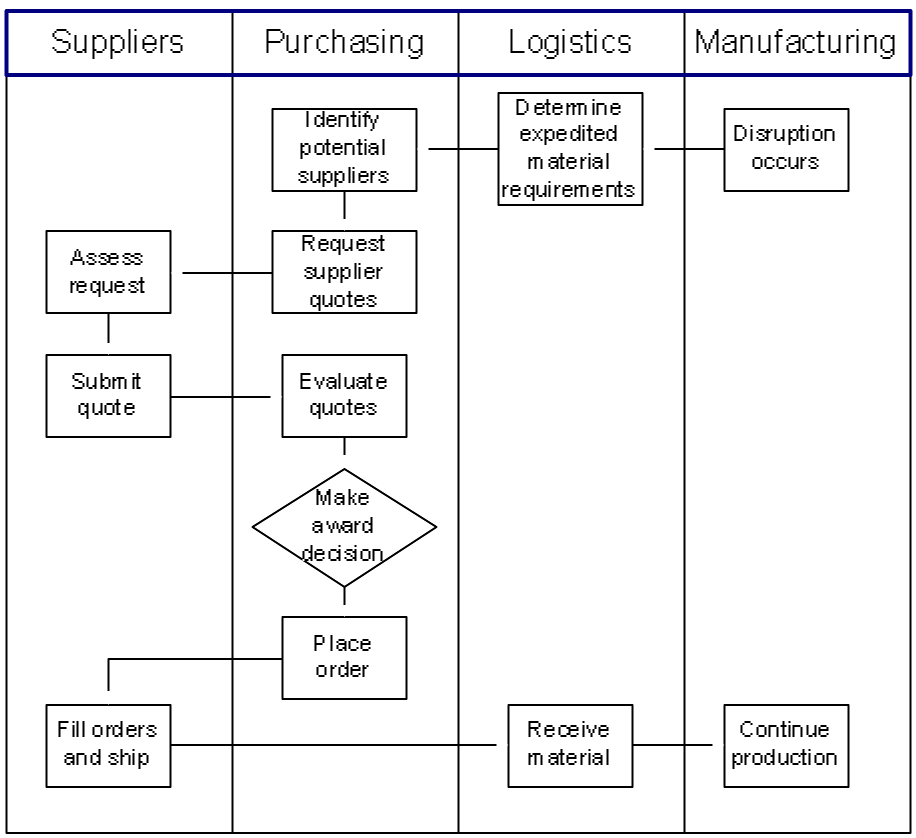
An important function in production organizations is the purchase of the goods and services needed to support the operations involved in fulfilling customer orders. A typical firm spends at least half of its revenues on external purchases of goods and services (Markham, Morales, and Slaight, 2000), so the selection of suppliers and the control of material flows on a day to day basis are central to the success of the enterprise. How well an organization manages its supply function influences all areas of performance – speed to market, quality, pricing and pricing flexibility, innovation, order management, and customer responsiveness (Burt, Dobler, and Starling, 2003). Information technology has played a role in this key management function through the EDI and XML technologies discussed earlier. This section describes the architecture and conceptual design of an e-Procurement System that fully exploits the real-time and dynamic characteristic of a service-oriented architecture (SOA) using Web services.

3.1 The e-Procurement Application Scenario

Purchasing goods and services in an organization can be viewed as two processes: supplier selection and the ongoing purchase of materials. Suppliers are selected in the sourcing process before production of a new product begins, and are static in that relatively few changes are made to the supplier base during the life of the product. This supplier selection process comprises defining the requirements for the purchased items, identifying potential suppliers, requesting supplier quotes, evaluating bids, and ultimately selecting suppliers based on expected performance as well as price.

Once suppliers are selected, the purchase of materials typically includes computing material requirements on an ongoing basis and then placing orders through either a blanket purchase order or a material release notification. Many of these ongoing purchases are routine, where a purchase requirement is specified early enough so that standard channels can be used, and where the selected suppliers are able to provide the required items in sufficient quantities and delivery timing. Today, the Internet supports both the supplier selection process through e-sourcing applications, as well as material purchases through EDI-enabled communication of material requirements to suppliers (Holloway and Higuera, 2002). For routine purchases, these processes have generally resulted in acceptable performance levels for the buying organization.

These processes don't work as well, however, for purchases that result from a disruption to the production process. Disruptions to production are caused by events such as changed or emergency orders, cancelled orders, customer requests for delivery that violate the normal lead time, excessively large error in sales forecasts, production gains or losses, and unsatisfactory quality in supplier materials. These events typically create a need for material that was not previously anticipated and so requires a special expedited purchase. In practice, production organizations incur extraordinary costs to recover from this type of disruption. It is for these expedited purchases that Web services play an especially important role, and so it is in this context that we explore the use of Web services.



there is a need for expedited material. In some cases, the inventory on hand is sufficient to cover the change in the production plan - in other cases, it is not. Suppose for example that a large customer order is cancelled, and in order to maintain production targets, different orders are pulled into the current production week. In this case, the components needed to produce the substituted customer orders may be substantively different from the inventory on hand and shipments en route, and so material will need to be expedited to support the revised production plan. In practice, the pre-selected suppliers are often unable to fill the expedited material request, and so managers scramble to find other qualified suppliers that can produce the required materials in time. When this cannot be accomplished, another wave of changes may be implemented in the plan. These changes drive "instability" in the manufacturer's production plan, and have been identified as detrimental to overall supply chain performance in a number of sources including Inman and Gonsalvez (2001).

For some components that are custom engineered for an application, only the pre-selected suppliers can fill material needs. But for commodity and standardized components, other qualified suppliers could fill the expedited material requests that result from production disruptions. A dynamic supplier identification and order placement process that is enabled by Web services would be helpful in these cases, as the purchasing organization can then quickly request and evaluate supplier quotes from potential suppliers. In practice, a Web Services Registry Database would provide information on potential suppliers who are known to be flexible enough to respond to expedited and emergency requests of this type. This expanded set of potential suppliers would then have an opportunity through the Web services enabled process to assess their ability to fill the special order and submit a quote if able. The process of requesting, assessing and submitting quotes needs to happen very quickly – in real time – for it to be useful to a disrupted production process. The purchasing organization then evaluates all quotes, makes an award decision, and places the order with the selected supplier(s). The supplier fills the order and ships the material, which is received by the logistics organization and provided to manufacturing to enable continued production. It is in this way that an ongoing material purchase process may be supported by the Web services technology.

3.2 An e-Procurement System Architecture

A prototype e-Procurement System using dynamic Web services are presented here to address the expedited purchases application presented in Section 3.1. The overall architecture design of the system is depicted in Figure 2. The system consists of several subsystems as discussed in the following.

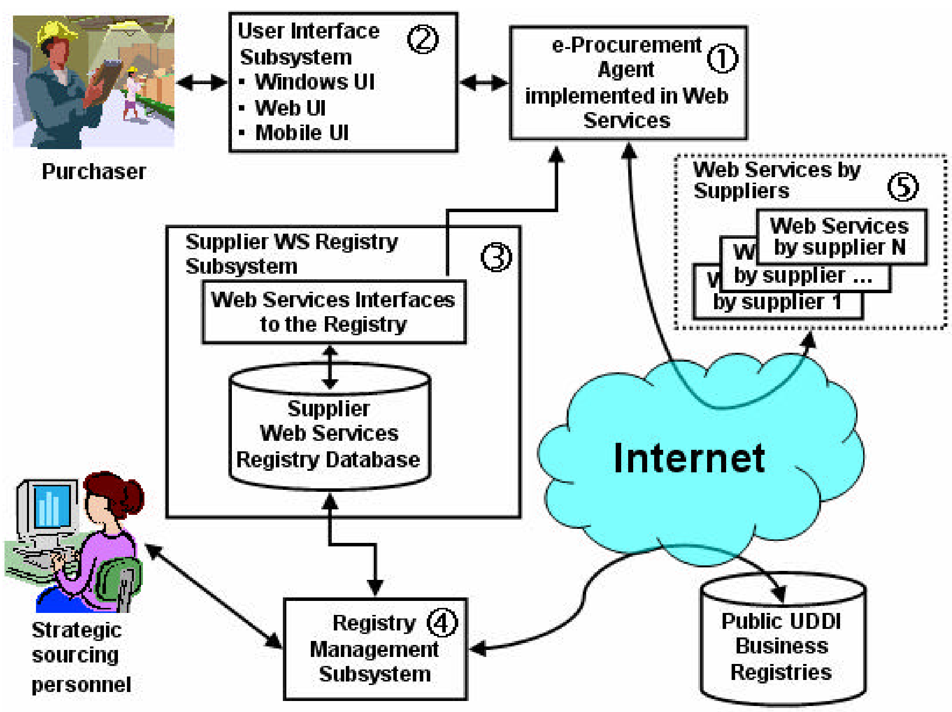


Figure 2 An architecture of a Web service-enabled e-Procurement system

1. e-Procurement Agent: The e-Procurement agent is a software component implemented in Web services. It serves as a middle-tier component to handle the interactions with the Web Services Registry and with the trading partners' Web services. It consumes Web services provided by the Supplier Web Services Registry and Web services provided by suppliers. The e-Procurement Agent is implemented as Web services to be consumed by the front-end user interface applications.
2. The User Interface (UI) Subsystem: Since the e-Procurement agent is implemented in Web services, the e-Procurement UI subsystem can be easily implemented using various user interface components that are capable of consuming Web services including:
    a. **A GUI client running on desktop computers.** GUI-based applications provide rich user interface elements and interactions. It is often used by purchasing personnel working from their desktop inside a company.
    b. **A browser-based application.** Web presentation layer components are web programs running on the server-side. These Web presentation layer programs can consume Web services and provide easy access to the e-Procurement system wherever there is Internet access.
    c. **Mobile clients.** Mobile device such as cell phones, personal digital assistants (PDAs) can consume Web services. Special micro-browsers can be used to get access a server-side Web program. The mobile web programs need to render Web pages in formats such as WML or cHTML that are appropriate for the requesting mobile devices.
3. Supplier Web Service Registry Subsystem: The public UDDI is too generic to be useful for the proposed e-Procurement System. It does not provide adequate information from an e-Procurement viewpoint. Therefore a private Web services registry is developed to store product and supplier information as well as Web addresses of Web Services Description Language (WSDL) (W3C, 2001a) files of Web services provided by all participating suppliers for various Web operations required for the e-Procurement application. The registry's database stores all qualified suppliers information including their capability, quality, as well as WSDLs for various interactions (e.g. price quote and order) require to complete an e-Procurement transaction. The Registry subsystem plays a key role in enabling the dynamic nature of the procurement process as it expands the list of potential suppliers, an important feature especially for the expedited purchases which the standard suppliers may not be able to fill. e-Procurement agent uses a set of Web operations published by the Registry Subsystem to access information of suppliers' Web services.
4. Registry Management Subsystem: Strategic sourcing personnel uses this subsystem to maintain the supplier data and their Web services entries. Potential suppliers are evaluated to determine whether these suppliers should be registered in the private Web Services Registry. New suppliers and their Web services entries can be continuously identified and updated by accessing the public Universal Directory, Discovery, and Integration (UDDI) Business Registries or directories of other public exchanges (UDDI.com, 2003).
5. Web Services by Suppliers: In our current design, suppliers who want to join the e-Procurement system need to implement Web services required by the e-Procurement system according a set of Web operations based specific defined by a published WSDL file. The internal implementations of these Web operations are insignificant. However, all suppliers need to support identical signatures for these Web operations, i.e. for the same operation from different suppliers, the operation name, the numbers of parameters and parameters' data types have to be the same. These Web services are considered as an extended and integral part of the e-Procurement system. Having these Web services available enables the supplier to "assess and submit quotes" in real time such that little or no human intervention is required to respond to a customer request.

As disruptions occur and items need to be acquired in short lead time, a Web service enabled e-Procurement system is able to quickly prepare a "quote request" so that a number of suppliers can respond with availability, price, and delivery, followed by a "quote evaluation" that awards and then transacts the order with the selected supplier(s). The e-Procurement System enables the buying organization to purchase supplies in real-time with close to fully automated transactions processing.

3.3 The e-Procurement System Design: A Three-Tier Approach

The e-Procurement agent is the key element of the e-Procurement system's architecture. It is a middle tier business logic component, implemented in Web services, which can be used to identify a set of qualified suppliers' information, to query for price and availability quotes, and to make a purchase recommendation. The design of the e-Procurement agent with emphasis on its interactions with other components of the system is illustrated in the following UML sequence diagram. A three-tier architecture approach consisting of the presentation layer, the business logic layer, and the data service layer, is taken in designing the e-Procurement system, consistent with the patterns of enterprise application architecture (Alur and Marks, 2001; Fowler, 2003; Microsoft, 2003). In the e-

Procurement System, a Windows-based and a mobile web user interface has been implemented as the **presentation layer** of the system. The e-Procurement agent is the **business logic layer** that encapsulates procurement business rules and hides the complexity of data service layer from the presentation layer component. The supplier-product data and WSDL entries provided by suppliers are packaged in the **data service layer**, i.e. the Web services registry. The registry functionality can be accessed by a set of public Web services operations which can be consumed by the e-Procurement agent.

This e-Procurement system can be modeled in UML as in Figure 3 where we show the material purchase process for expedited requirements, as it would execute in a Web services enhanced purchasing environment. The sequence diagram indicates a series of operations performed by a purchaser to procure components, materials or services from suppliers. With UML, Web services can be modeled as classes using the UML diagram. In this sequence diagram, a Web service is modeled as an object (i.e. an instance of a Web service class). The Web methods or Web operations that specify the interfaces provided by Web services can be modeled as messages. Web operation signatures consist of method names, parameters and their data types of class operations. An iteration (loop) in the sequence diagram is represented by an asterisk (*) before the operation signature. So for example, a condition such as *[for each WSDL in WSDLs] can be added in front of the operation' s name (Miller, 2001).

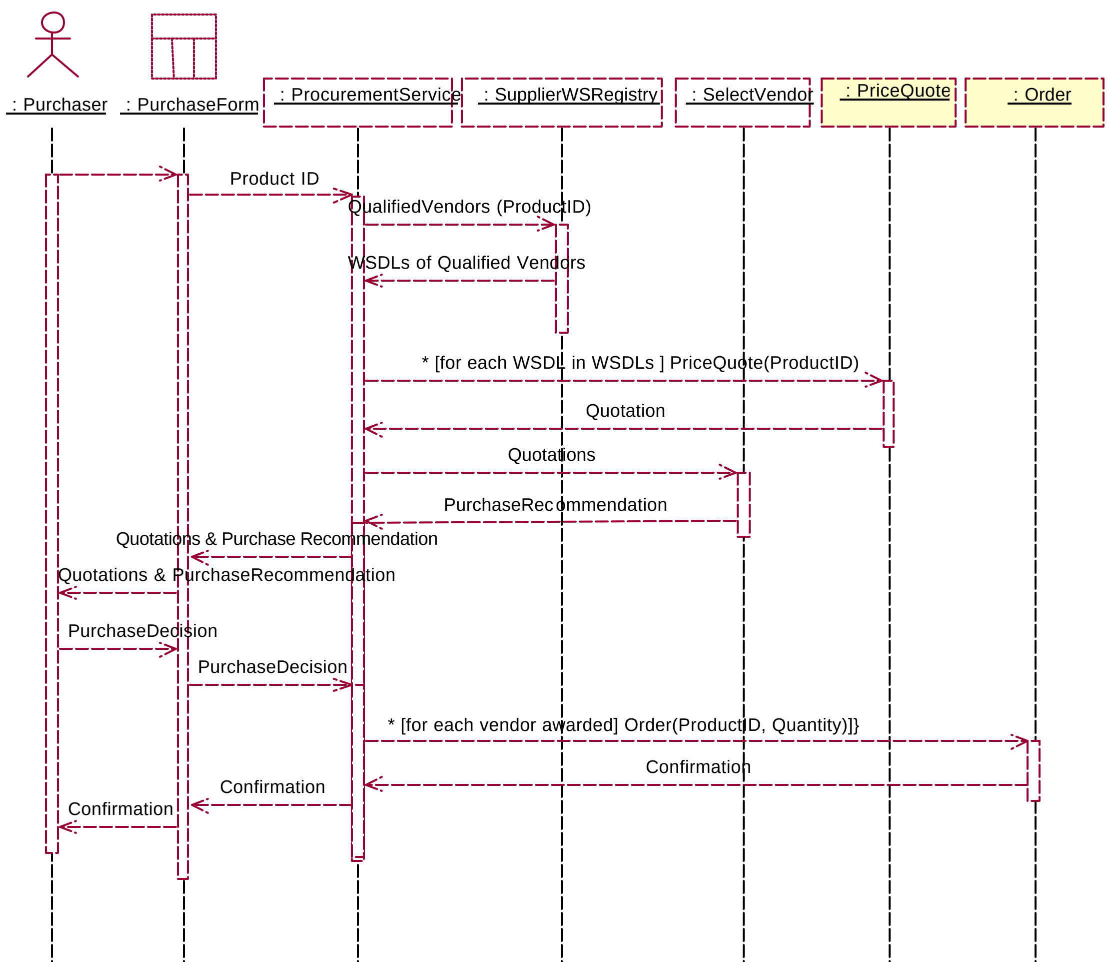


Figure 3 The UML sequence diagram for real-time procurement

The UML sequence diagram represents a typical scenario of the e-Procurement systematic and real-time purchasing process. A purchasing officer is alerted by the material control system that an inventory shortage has or is about to occur, and logs onto the e-Procurement system. The product ID is entered to the PurchaseForm presentation layer component. The productID submitted by the end user via the PurchaseForm will be submitted to the Purchase Service (i.e. the e-Procurement Agent depicted in Figure 2). The Purchase Service component that is implemented as Web services will call the Web services provided by the Web Services Registry to retrieve WSDLs of all qualified suppliers for the requested product. Thus we assume the supplier qualification process is independent and completed in advance, perhaps continuously in anticipation of material shortages. An iteration of a PriceQuote Web service call is made to each of the qualified suppliers. The supplier's quotation in XML format will be returned and aggregated by the Purchase Service agent.

Potentially all the quotations can be sent to a purchase agent who applies some established business rules to recommend the purchasing award decision. The business rule for the award decision may be as simple as "buy from the supplier with the lowest quoted price first" or "buy from the supplier with the fastest delivery time", or a combination of additional factors such as supplier quality and historic delivery performance. For example, total cost of ownership (TCO) may be used to score the suppliers who are able to provide the needed components on time as an input to the allocation of the order if multiple suppliers are to be used. TCO is a best practice that incorporates product price, the buying firm's purchasing-related costs, and expected supplier performance into the decision process. The performance metrics may include reliability costs, quality costs, asset utilization, downtime and cycle time, and inventories as affected by the supplier (Ellram and Siferd, 1993; Ellram, 1995). The data used to compute an award decision then be made more dynamic and based on real time data, yet consistent with the firm's purchasing and materials management strategies.

The quotations or the recommended purchasing decision will be shown to the purchasing officer via the PurchaseForm component. Once a purchase decision has been made, purchase orders will be sent to the selected suppliers via Web services. The "Order" web operation's WSDL of the supplier is retrieved from the Web Service Registry dynamically to complete this task in this e-Procurement process.

4 Implementation of the Dynamic e-Procurement System

With the advent of Web services, an e-Procurement system enabled by Web services can be used to facilitate the purchasing of expedited purchases where several qualified suppliers are able to supply the required product. The e-procurement process presented here takes advantage of the dynamic and real-time capabilities of Web services. We depict an illustrative example, and a prototype that portrays how a buyer might readily access and make a decision concerning the e-procurement of material for a production facility in the B2B segments of a supply chain. This modified procurement process enhances the performance of the supply chain, making the supply chain more responsive to customer needs and changes to those needs, as well as internal disruptions that occur with regularity within a production operation.

4.1 The Design Web Services Registry Database and Web Service Interfaces

In our implementation, information is stored in a relational database, and a set of Web services are implemented to provide interfaces so that other subsystems or applications can get access to the information stored in the Registry database. Part of the Registry database design represented in entity relationship diagram is illustrated in Figure 4.

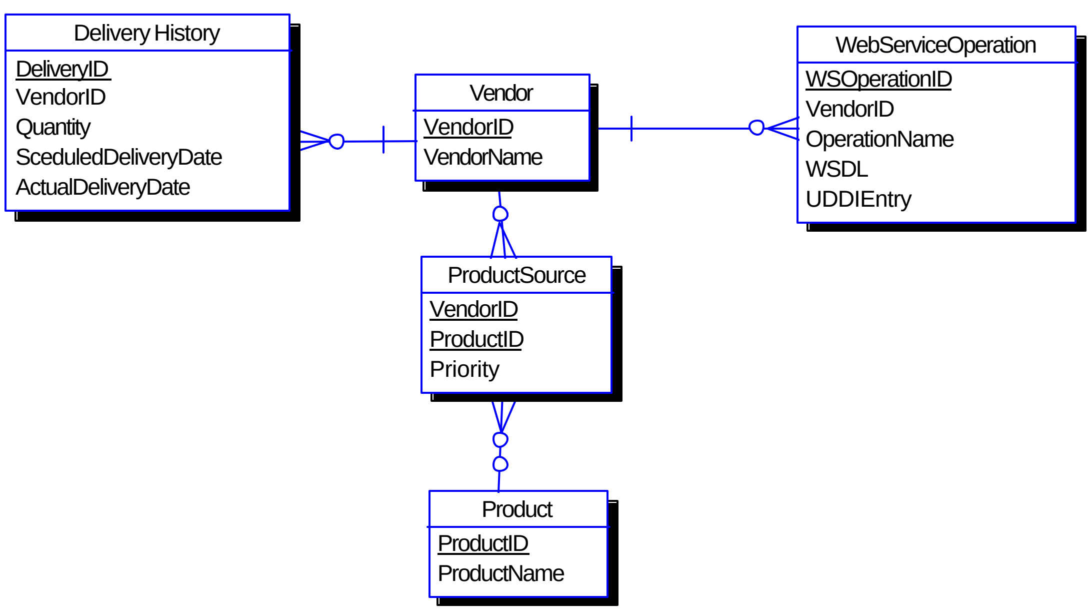


Figure 4 Part of the data model for the Supplier Web Services Registry database

The delivery history entity type is derived from the historical purchasing and delivery transaction data and can be used for determining each supplier's ranking or rating in a dynamic supplier selection process, or in allocating an expedited order to more than one supplier. Although we use delivery history only in this implementation, other metrics such as product quality, which is based on scrap and returns history, could be similarly incorporated into the process. The ProductSource is an associative entity type that provides a cross-reference table indicating what products can be provided by suppliers. The WebServiceOperation entity type stores information for each Web service operation provided by all qualified suppliers. The WSDL entry to access the Web service operation is recorded here. An optional UDDI entry attribute is used to store the corresponding UDDI entry of the Web service for those suppliers who register their Web services at public UDDIs. This UDDI entry will be used to update the WSDL entry when the existing WSDL entry has been become outdated.

This data model supports and is consistent with the practice of using TCO for supplier scoring and order allocation as described earlier. In the data model, the priority attribute in the ProductSource table is used to assign priority or weight in allocating orders to suppliers. The priority value associated with each product ID and supplier combination is based on delivery reliability and product quality performance derived from delivery history data. The priority attribute may be used to rank suppliers, so that in the event that the first choice supplier cannot provide the full order, a second supplier is readily identified to ship either the full or remaining quantity. Alternatively, the suppliers may be assigned an allocation of orders based on historical performance as is often done with standard purchases. For example, a favored supplier may be allocated 75% of the ongoing cumulative purchase decision, while the secondary supplier receives 25%. The priority attribute may be structured to support this.

Major Web services operations developed to provide interfaces to the e-Procurement Web service Registry are:

1. GetSupplierWSDL (SupplierID, WebOperation): The WebOperations are predefined set of operations implemented by all suppliers in the Registry. For example, "PriceQuote" operation allows us to send a request for a price quote for a material or component and receive the quotation from the supplier. "Order" is the operation allows a Web service consumer to order a product from a supplier via a Web service interface. The return value of this Web method is the WSDL of a supplier (identified by SupplierID that implements the WebOpration.
2. SetSupplierWSDL (SupplierID, WebOperation, WSDL): This method is used to add or update the WSDL entry of a supplier's web operation.
3. QualifiedSupplier (ProductID): Retrieve an array of suppliers' supplierIDs based on the ProductID.
4. PlaceOrder (ProductID, Quantity, SupplierID): Place an order for a certain Quantity of a product with the ProductID from a supplier with the SupplierID. The "Order" Web operation from a Web service of the target supplier will be invoked to complete this request.

The buyers and sellers in a supply chain need to come together to define a set of standard Web services operations with the same operation signatures (i.e. operation name, same number of parameters, and same data types for each parameter). This standard effort is critical for the success of developing this dynamic procurement system and resources and time should be allocated to ensure all suppliers are informed and agree to such implementation.

In the near term, we believe that human intervention is required to screen suppliers. Currently, the function of managing the Registry is handled by product sourcing personnel using the Registry Management Subsystem. In the long term, a vertical UDDI operated by eMarketplace operators may guarantee the quality of suppliers listed in the UDDI registry. The e-Procurement database can then be linked to public UDDI to constantly update the WSDLs for each supplier and searching for new suppliers who are qualified to provide us with products required for the company's production process. Supplier performance may be continually assessed using both internal and external data and updated automatically. Web services allow for sharing performance data concerning specific suppliers, and possibly accomplishing the quality assessment itself.

4.2 The Implementation of the e-Procurement Prototype

A Registry implemented in the relational database contains Web services entries for various Web service operations required for interacting between the buyers and suppliers in this e-Procurement system. A set of Web services is implemented to provide interfaces for other software applications and components to interact with the Web services Registry. Mapping between buyer's Product number and the product code used by different suppliers are stored in the Registry so that the translation of internal product numbers to supplier's product numbers is hidden from the Web services consumers.

A three-tier architecture is deployed in our design and implementation. The Presentation layer can be a traditional Windows interfaces, Web user interface, or Mobile Web application. This can be achieved easily due to the Web services based implementation of business layer and data services layer components which can be easily consumers by clients using traditional clients or smart clients. A Purchase Service software component is implemented in Web services served as a business layer component. The Purchase Service can be viewed as a purchasing agent.

The prototype e-Procurement system is implemented in Visual Studio .NET Professional 2003 under Microsoft Windows 2003 Server. The implementation tool and platform are chosen because of the strong support for Web services development in the .NET Framework environment. The Windows-based presentation layer is implemented as illustrated in Figure 5.

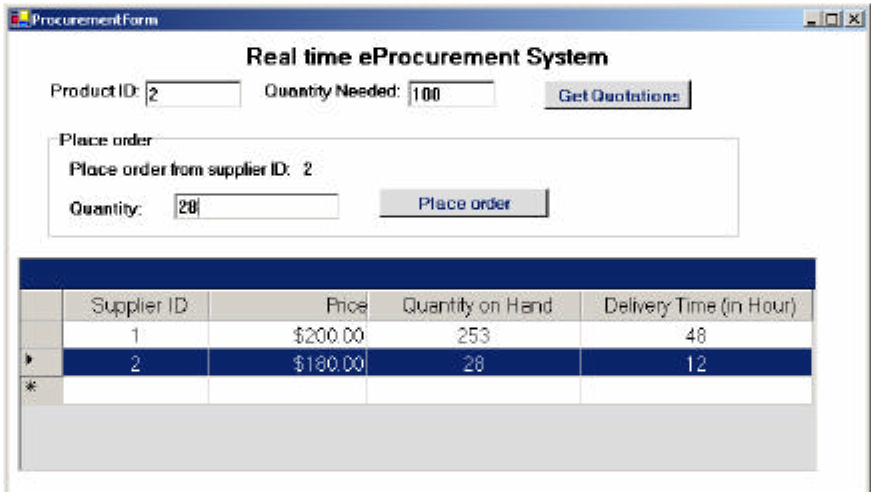


Figure 5 The Windows-based front end of the e-Procurement System

The product ID of the out-of-stock inventory and the quantity needed may be generated by the internal ERP system. It is a standard practice for inventory personnel to modify the amount to be purchased based on additional soft data and judgment. The user may manually enter a product number directly for a product that is out of the stock and need to be replenished, and then she can click the "Get Quotations" button to send the get quotation request to qualifying suppliers for the requested products. The quotation responses from suppliers include the following information: supplier ID, price quote, quantity on hand at the supplier, and estimated shipping time in hours.

The user can click on the heading of each column in the DataGrid to sort the results by that attribute in ascending or descending order to evaluate these quotations based on different attributes. An additional call to a business rule engine applying various business rules in making recommended purchase can be developed. The user can select a quotation by clicking a row in the DataGrid corresponding to a specific supplier' s quotation. The user will then enter the quantity to be ordered from the supplier before clicking the "Place Order" button to issue an online order via an order Web service provided by the selected supplier. The binding to the selected supplier with whom to place the order is done dynamically.

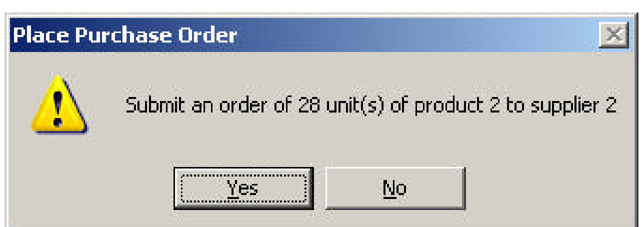


Figure 6 Confirmation of real-time order placement

A confirmation dialog box will appear before submitting an order. The result of the order submission will be displayed back to the user. A purchase order will be created internally as an auditing trail and for tracking the purchase.

Mobile computing devices that are capable of consuming Web services can use distributed components implemented as Web services to get access to legacy data and applications (Wigley and Roxburgh, 2002). This approach will extend the accessibility of our e-Procurement system. This mobile application can be used by logistics manager checking inventory or by sale persons visiting clients. The greater mobility provided by the system allows the company to initiate the procurement process at anytime and anywhere by appropriate personnel.

4.3 Implementation of the Purchase Agent

The XML Schema DsPriceQuoteResult listed in Appendix A is used for passing aggregated quotation results back to the client of the Web operation GetPriceQuote of the Procurement Service. The code for the Web method GetPriceQuotes can be found at Appendix B. In order to provide dynamic binding of PriceQuote Web service from qualified suppliers based on the search Supplier Registry, we have to modify the generated Web service proxy (referred to as PriceQuoteWS at Line 13 in Appendix B) generated by the system for the PriceQuote Web service. The generated Web service proxy by default uses static binding. A new constructor for the proxy call is added manually to accept a new WSDL reference as a parameter. The new constructor is used to instantiate an instance of the PriceQuote Web service based on a parameter representing the WSDL for the Web service provided by a supplier (Line 13 in Appendix B). The code from line 12 ~ 27 is an iteration construct to invoke the GetQuotation operation of the PriceQuote Web service for each of the qualifying supplier dynamically to request price quotations from them for a specific product. The aggregated quotations returned from suppliers are stored in an XML document referred to as DsPriceQuoteResult.

4.4 Performance Evaluation and Improvement

Currently the dynamic Web services calls are performed synchronously with very good response time. In a simulation study, we obtained 3 seconds response time for calling 100 suppliers PriceQuote Web services. In the simulated environment, all supplier Web services are running on the same server machine so network traffic is not factored into the performance measurement. The server machine is Dell Dimension 2350 with an Intel Celeron 2.00 GHz CPU and 630 mega bytes main memory. The performance may be further improved if we implemented the Web services call using asynchronous Web services calls with callback functions and multithreading. This supports concurrent processing such that the delay in some service calls may not have major effect on the overall performance of the system (Freeman and Jones, 2003).

Even though we have explored here the short-term expedited purchase process, the general architecture and design of the e-Procurement system is still appropriate for supply processes with long-term transactions such as the routine purchase. The technical architecture of the system should be based on a message queue-based technology. This approach will give our system more loosely coupled design such that the e-Procurement system does not have to reply on a request-response model of communications. Potentially, the Web services definitions provided by all suppliers don' t have to be the same; however, additional data mapping may be needed. This greatly increases the complexity of the implementation efforts of all trading parties involved (Fowler, 2003).

5 Impact of Web Services on e-Procurement and Adoption Issues

Web services are language, operating system, and platform independent. Therefore, all suppliers can potentially join the e-Procurement System by providing a set of Web operations according the WSDLs set by the e-Procurement System to expose their price quotation function and ordering function and register them in the Web Services Registry. Collaborations among Web Services are bound at runtime at the initiation of a new purchase. Dynamic service discovery via the Registry and invocation via the e-Procurement agent enables just-in-time integration of the e-procurement process. Since multiple suppliers are used as sources of registered products plus the automation of the price quotation and ordering, the expedited purchase process can provide the buyers with real-time price and availability information of products from qualified suppliers and improve a broad set of performance of business processes within a supply chain. We have explored here its unique abilities to improve the procurement operation, and demonstrated how the procurement process can reflect the availability of continually updated information as well as the real time processing of special purchase orders between trading partners in the supply chain.

Earlier research has described the technical features of Web services and also discussed how Web services in particular can be used to enhance firm performance (Samtani and Sadhwani, 2002b). Samtani and Sadhwani (2002b) argued that Web services are adopted because of internal financial and technical advantages, which is realized through their effect on internal operations. Financially, adopting Web services reduces costs, increases opportunities for new customers who require the use Web services to conduct business online with them, and potentially drives a revenue stream where the application developed as a set of Web services might serve other customers willing to pay for the service.

Web services may also improve integration among heterogeneous and distributed systems; hence, enhance business process integration. There are likely to be additional benefits, then, associated with Web services in this extended enterprise context. These can be measured with two general criteria, externally focused customer satisfaction and internally focused cost efficiencies, consistent with the set of attributes proposed in the SCOR model (Supply-Chain Council, 2003). The SCOR (Supply Chain Operations Reference) model provides a taxonomy and a structure for business process improvement across five functional elements: plan, source, make, deliver, and return. The model categorizes performance into five metric categories: cost, asset management, reliability, responsiveness, and flexibility. Performance on each of these attributes may improve with implementation of Web services in the expedited purchase business process. In the illustration provided in this paper, responsiveness to a disruption in a customer's production process is emphasized.

Practically speaking, little research has been done concerning the impact of Web services on the procurement operation, and this remains an important future area of research. Earlier research has however explored the benefits of EDI in the purchasing function and shows that production and transaction costs are reduced, as in Sriram and Banerjee (1994). When extended to the Web exchanges, a decrease in price may occur as new suppliers are found on the Internet, and operational improvements may result from re-designed workflows (Kaplan and Sawhney, 2000). Asset utilization relative to inventory also may be expected to improve as information typically substitutes for inventory in a supply chain (Gavirneni, Kapuscinski, and Tayur, 1999; Kekre, Mukhopadhyay, and Srinivasan, 1999; Lee, So, and Tang, 2000).

Externally focused customer satisfaction may also be improved with Web services. For example, reliability has been shown to improve with IT enhanced information flows in the supply chain, in part because more information is available to suppliers as disruptions unfold (Ahmad and Schroeder, 2001; Walton and Marucheck, 1997). This reliability improvement may be greater with Web services because the supplier set is extended beyond the same supplier(s) who won the initial bid for standard purchases based on a stable disruption-free production process. Responsiveness in terms of order response time may also improve as information is made more generally available in supply chains (Cachon and Fisher, 2000). Build-to-stock manufacturers are likely to experience improvement in customer response because finished inventory is more consistently available and disruptive shortages are better managed. Likewise, build-to-order manufacturers are likely to improve delivery performance because component inventories are more readily replenished when a disruption to the process occurs and an expedited order for components is placed through a Web services enabled e-procurement system.

There are several issues that need to be addressed before this Web services-based process integration gains momentum:

- Business-level standards need to be developed to provide consistency of the Web services registered in the UDDI. Industry-specific Web services standards for business transactions or documentation (e.g., OASIS Universal Business Language) may emerge.
- Security is the main problem facing Web Services and a major concern of many potential Web services adopters. Web Services do not have widely adopted security protocols and standards dedicated for Web services. Many organizations use existing security measures such as HTTPS or encrypt the whole SOAP messages to secure Web services communications. Dedicated standards such as Security Assertion Markup Language (SAML) are under development.
- An additional concern with the use of Web services in an operating environment is the quality of service, measurable in terms of reliability, performance, and availability of the Web service itself. Here, the overall reliability of the Internet impacts the reliability of Web services, along with the uptime performance of the application servers where the applications reside. A promising solution to this problem is the use of Web services grids, where Web services are built on of computer grids (Tan, Topol, Vellanki and Xing, 2003).

New development tools and platforms supporting Web services are abundant and they are very helpful in shortening the learning curve. Web services have a relatively smooth learning curve for developers who have 3-tier architecture design and programming experience. The cost of adopting Web services is marginal. This is particularly the case for enterprises that already have an e-business infrastructure in place. Once supply chain applications using Web services have proven to be beneficial to business as illustrated in this paper, companies may seriously consider building more inter-organizational information systems to further strengthen the integration of their supply chain.

6 Conclusions

We contribute to the development of Web services in supply chain applications in this work by first arguing that earlier information technologies don't support the real time and dynamic needs of many supply chain processes. We demonstrate that Web services can provide this support with a prototype that illustrates the capabilities of Web services. For the expedited purchase process, we find that the dynamic nature of Web services lets supplier identification be continually updated as suppliers drop in and out of the otherwise static supplier list. Additionally, varying and dynamic criteria for awarding an order can be used given broad access to internal and external supplier registries through Web services. Additionally, we summarize research addressing benefits gained from earlier information technologies, and we see that a broad set of benefits are likely to be realized in the supply chain because of the enhancements afforded by Web services. Certainly, empirical research is needed to better understand these benefits, along with research that addresses the barriers to implementation and the development of business processes that take specific advantage of Web services unique characteristics.

A number of Web services research issues and potential extensions to the e-Procurement System have been identified and are discussed in the following:

- The e-Procurement System prototype uses a private exchange to support the procurement process. Additional Web services interfaces can be developed to allow suppliers to register their products and Web services via a browser or via Application Program Interfaces (API) implemented as Web services, so that the maintenance of the Supplier Web Services Registry can be initiated by suppliers directly.
- Further extending the product and supplier information with buyer profile information, demand information and customer sales data that has been acquired through a query using secure Web services can support the collaborative demand forecasting process. We are developing such extension to turn the system into a public B2B exchange. Instead of using XML data to maintain a product catalog with periodic pricing updates, only product data and Web services entries for real-time price quotation and ordering are required.
- Current acceptance of UDDI in real commerce has been slow in part due to the lack of domain-specific knowledge, such as product data. This paper proposes an alternative design and implementation complementing current UDDI registry functions. It may accelerate the development of industry specific vertical Web services registries to support industry specific vertical marketplaces (Anderson and Lee, 2001). Adapting existing standards such as Rosetta's PIP - a business processes interface standard between trading partners - can speed up the development of such industry specific Web services standards (RosettaNet.org, 2003).

Likewise, several operations management research issues relative to Web services have emerged. Even though a great deal of earlier research has addressed best procurement practices, many of the questions should be revisited as processes may be re-engineered as is done here to take specific advantage of Web services characteristics. The use of advanced technologies such as Data Envelopment Analysis and best practices such as Total Cost of Ownership for selecting suppliers and possibly for allocating purchase quantities become more viable as Web services provide additional data and processing capabilities. Also, the impact of Web services on specific attributes of supply chain performance needs to be further investigated as well. In particular, little research exists on the issue of flexibility in the supply chain due to information technologies in the procurement processes.

The function of expedited purchases explored here is a relatively straightforward application relative to Web services. It is important in future research to also investigate supply chain integration activities that address emerging cross-enterprise integration functionality. These include the use of Web services in automated replenishment systems, as well as in collaborative forecasting as mentioned earlier. Opportunities for the use of Web services also exist in collaborative production planning and scheduling across enterprises in the supply chain, as well as cross-enterprise project planning and management. Another research area that merits investigation is the integration of procurement in the buying firm with supplier side dynamic pricing which advocates quote formation based on available capacity and other factors. In this way, suppliers may allocate their production capacity across multiple customers while customers may allocate their orders to multiple suppliers in an optimal fashion.

We have argued in this paper that deploying Web services reduces integration costs and improves the timeliness of services to a degree that would otherwise not be achievable in an extended enterprise. Suppliers can better respond to customer demands for expedited materials with the dynamic and real time capabilities of Web services. The expected benefits to building Web services into supply chain processes include both cost and responsiveness, so impact can be expected to be realized in a variety of organizations with a full range of business strategies. Certainly, additional research is needed to further explore these opportunities and challenges.

REFERENCES

Adams, H., et al., "Custom Extended Enterprise Exposed Business Services Application Pattern Scenario," http://www-106.ibm.com/developerworks /webservices/library/ws-best5/, Jan. 1, 2003

Ahmad, S. and R. G. Schroeder, "The Impact of Electronic Data Interchange on Delivery Performance," *Production and Operations Management*, 10:1, 16-30, 2001.

Alur, D., J. Crupi, and D. Marks. Core J2EE Patterns: Best Practices and Design Strategies. Prentice-Hall, 2001.

Anderson D. and H. Lee, "New Supply Chain Business Models - The Opportunities and Challenges," *Ascet: Achieving Supply Chain Excellence Through Technology*, 3, 12-18, April 15, 2001.

Burt, D. N., D. W. Dobler, and S. L. Starling, *World Class Supply Management: The Key to Supply Chain Management*, McGraw-Hill Irwin, Boston, Massachusetts, 2003.

Cachon, G. P. and M. Fisher, "Supply Chain Inventory Management and the Value of Information," *Management Science*, 46:8, 1032-1048, 2000.

Camm, J. D., T. E. Chorman, F. A. Dill, J. R. Evans, D. J. Sweeney, and G. W. Wegryn, "Blending OR/MS, Judgment and GIS: Restructuring P&G's Supply Chain," *Interfaces*, 27:1, 128-142, 1997.

Chen, M., **"**Factors Affecting the Adoption and Diffusion of XML and Web Services Standards for E-business Systems**,"** *International Journal of Human-Computer Studies*, 58:3, 259-279, March 2003.

Chopra, S. and P. Meindl, *Supply Chain Management: Strategy, Planning and Operations*, 2nd edition, Pearson Prentice-Hall, Upper Saddle River, New Jersey, 2004.

CPRF.ORG, CPFR Voluntary Guidelines, version 2.0, http://www.cpfr.org/documents /pdf/CPFR_Tab_7.pdf, June 2002.

Davis, D. "Third Parties Deliver," *Manufacturing Systems*, 13, 66-68, 1995.

Ellram, L. M., "Total Cost of Ownership: An Analysis Approach for Purchasing," *International Journal of Physical Distribution and Logistics*, 25:8, 4-23, 1995.

Ellram, L. M. and S. P. Siferd, "Purchasing: The Cornerstone of the Total Cost of Ownership Concept," *Journal of Business Logistics*, 14:1, 163-184, 1993.

Freeman, A. and A. Jones, "Chapter 15: Consuming XML Web Services Asynchronously," Microsoft .NET XML Web Services Step by Step, Microsoft Press, 333-362, 2003.

Fowler, M., Patterns of Enterprise Application Architecture. Addison-Wesley, 2003.

Fu, S., etc., "A Practical Approach to Web-Based Internet EDI," IBM, http://www.research.ibm.com/ iac/papers/icdcsws99.pdf , May 1999.

Gavirneni, S., R. Kapuscinski, and S. Tayur, "Value of Information in Capacitated Supply Chains," *Management Science*, 45:1, 16-24, 1999.

Holloway, C. A. and A. Higuera, "e-Sourcing at Sun, version A," Case Number OIT-34, Stanford Graduate School of Business, 2002.

Inman, R. R. and D. J. A. Gonsalvez, "The Causes of Schedule Instability in an Automotive Supply Chain," *Production and Inventory Management Journal*, 38(2): 26-31, June 1, 2001.

Hahn, C. K., E. A. Duplaga, and J. L. Hartley, "Supply-Chain Synchronization: Lessons from Hyundai Motor Company," *Interfaces*, 30:4, 32-45, 2000.

Jouenne, T., Henkel-Eroski CPFR Pilot Case Study, http://www.cpfr.org/HenkelEroski.pdf, 2000.

Kaplan, S. and M. Sawhney, 2000, "E-Hubs: The New B2B Marketplaces," *Harvard Business Review*, May-June 2000.

Kekre, S., T. Mukhopadhyay, and K. Srinivasan, "Modeling Impacts of Electronic Data Interchange Technology," in Quantitative Models for Supply Chain Management, S. Tayur, R. E. Ganeshan, and M. J. Magazine, Eds. Boston: Kluwer Academic Publishers, 1999.

Koloszyc, G., "Retailers, Suppliers Push Joint Sales Forecasting," *Stores*, June 1998.

Lee, H. L., K. C. So, and C. S. Tang, "The Value of Information Sharing in a Two-Level Supply Chain," *Management Science*, 46:5, 626-643, 2000.

Lin, G., M. Ettl, S. Buckley, S. Bagchi, D. D. Yao, B. L. Naccarato, R. Allan, K. Kim, and L. Koenig, "Extended-Enterprise Supply-Chain Management at IBM Personal Systems Group and Other Divisions," *Interfaces*, 30:1, 7-25, 2000.

Markham, W. J., J. T. Morales, and T. H. Slaight, "Creating Supply Advantage by Leveraging the Strategic Nature of Procurement", in The Purchasing Handbook, 6th edition, J. Cavinato and R. Kauffman, Eds. McGraw Hill Publishing, New York, NY, 2000

Microsoft, Application Architecture for .NET: Designing Applications and Services, Microsoft Press, 2003.

Miller, G., "Conditional Logic in Sequence Diagrams," http://www-106.ibm.com/developerworks/ java/library/j-jmod0605/

Mitchell, R.L. "Unilever Crosses the Data Streams," *Computerworld*, December 17, 2001.

Robins, G. "Pushing the limits of VMI," *Stores*, pp 42-44, 1995.

RosettaNet.org, "RosettaNet Partner Interface Processes", http://www.rosettanet.org/standards, Sept. 2003.

Samtani, G. and D. Sadhwani, "Enterprise Application Integration and Web Services," in Web Services Business Strategies and Architectures, P. Fletcher and M. Waterhouse, Eds. Birmingham, UK: Expert Press, LTD, pp. 39-54, 2002a.

Samtani, G. and D. Sadhwani, "Return on Investment and Web Services," in Web Services Business Strategies and Architectures, P. Fletcher and M. Waterhouse, Eds. Birmingham, UK: Expert Press, LTD, pp. 9-24, 2002b.

SAX Project, About SAX, at http://www.saxproject.org/, June 2002.

Stencil Group, Defining Web Services, http://www.stencilgroup.com/ideas_scope_ 200106wsdefined.pdf, 2001.

Sriram, V. and S. Banerjee, "Electronic Data Interchange: Does Its Adoption Change Purchasing Policies and Procedures?" *International Journal of Purchasing and Materials Management*, 30:1, 31-40, 1994.

Supply-Chain Council, Supply-Chain Operations Reference Model, version 6.0, June 10, 2003.

Tan, Y. S., B. Topol, V. Vellanki, and J. Xing, Business Service Grid, Part 1: Introduction, http://www-106.ibm.com/developerworks /ibm/library/i-servicegrid/, Feb 2003.

Turban, E., etc., *Electronic Commerce: A Managerial Perspective*, Prentice Hall. 2002.

UDDI.com, About UDDI, http://www.uddi.org/about.html, 2003.

Varon, E. “What you need to know about public and private exchanges,” *CIO Magazine*, September 1 2001.

Vecchio, D. “Legacy Software: Junkyard Wars for Web Services?” Gartner Symposium/ITxpo, Orlando, FL, USA, 2001.

W3C, Simple Object Access Protocol (SOAP) 1.1, W3C , http://www.w3.org/TR/SOAP/, May 8 2000a.

W3C, Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, http://www.w3.org /TR/REC-xml, October 6, 2000b.

W3C, Document Object Model (DOM) Level 2 Core Specification, Version 1.0, W3C Recommendation at http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113/13 November, 2000c.

W3C, Web Services Description Language (WSDL) 1.1, W3C , http://www.w3.org/TR/wsdl, March 15, 2001a.

W3C, XML Schema Part 0: Primer, W3C Recommendation, http://www.w3.org/TR/xmlschema-0/, May 2, 2001b.

W3C, Extensible Stylesheet Language (XSL) Version 1.0, W3C Recommendation, http://www.w3.org/TR/xsl/, October 15, 2001c.

Walton, S. V. and A. S. Marucheck, “The Relationship Between EDI and Supplier Reliability,” *International Journal of Purchasing and Materials Management*, pp 30-35, Summer 1997.

Wigley, A. and P. Roxburgh, *Building .NET Applications for Mobile Devices*, Microsoft Press, 2002.

Zilbert, A B., “A Comparative Study of Traditional Electronic Data Interchange versus Internet Electronic Data Interchange,” in D Colton, J Caouette, and B Raggad (Eds.), Proceedings ISECON, v 17 (Philadelphia): 501. AITP Foundation for Information Technology Education and at http://colton.byuh.edu/isecon/2000/501/, 2000

Appendix A: The XML Schema for Price Quotation Result

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element msdata:IsDataSet="true" name="DsPriceQuoteResult">
  <xs:complexType>
   <xs:choice maxOccurs="unbounded">
    <xs:element name="PriceQuoteResult">
     <xs:complexType>
      <xs:sequence>
       <xs:element minOccurs="0" name="SupID" type="xs:long" />
       <xs:element minOccurs="0" name="Price" type="xs:decimal" />
       <xs:element minOccurs="0" name="QuantityOnHand" type="xs:int" />
       <xs:element minOccurs="0" name="DeliveryTime" type="xs:float" />
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:choice>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Appendix B: The Code for Dynamic Web service Operation for Getting Multiple Price Quotations

```
<WebMethod()> Public Function GetPriceQuotes _
  (ByVal ProductID As Integer)  As DsPriceQuoteResult
Dim MySupplierRegistry As New SupplierRegistry.SupplierQuery
Dim MyPriceQuoteResult1 As New DsPriceQuoteResult
Dim DsPriceQuoteWSDL1 As New SupplierRegistry.DsPriceQuoteWSDL
Dim SupplierWSDL As _
 SupplierRegistry.DsPriceQuoteWSDL.PriceQuoteWSDLRow
Dim SupplierWSDL As String
Dim MyPriceQuoteWS As PriceQuoteWS.WSQuery
Dim MyPriceQuoteDs As PriceQuoteWS.DsPriceQuote
Dim drQuoteResult As DsPriceQuoteResult.PriceQuoteResultRow
Dim dtQuoteResult As New DsPriceQuoteResult.PriceQuoteResultDataTable
DsPriceQuoteWSDL1 = MySupplierRegistry.GetPriceQuoteWSDLs(ProductID)
For Each SupplierWSDL In DsPriceQuoteWSDL1.Tables(0).Rows
  MyPriceQuoteWS = New PriceQuoteWS.WSQuery(SupplierWSDL.WSDL)
  MyPriceQuoteDs = MyPriceQuoteWS.GetQuotation(SupplierWSDL.supPN)
  Dim MyPriceQuoteDt As _
  PriceQuoteWS.DsPriceQuote.PriceQuoteDataTable
  MyPriceQuoteDt = MyPriceQuoteDs.Tables(0)
  Dim MyPriceQuoteDr As PriceQuoteWS.DsPriceQuote.PriceQuoteRow
  If Not (MyPriceQuoteDt Is Nothing) Then
    drQuoteResult =  _
MyPriceQuoteResult1.PriceQuoteResult.NewPriceQuoteResultRow
    MyPriceQuoteDr = MyPriceQuoteDt.Rows(0)
    drQuoteResult.SupID = SupplierWSDL.SupID
    drQuoteResult.Price = MyPriceQuoteDr.Price
    drQuoteResult.DeliveryTime = MyPriceQuoteDr.DeliveryTime
    drQuoteResult.QuantityOnHand = MyPriceQuoteDr.QuantityOnHand
    MyPriceQuoteResult1.PriceQuoteResult.Rows.Add(drQuoteResult)
  End If
Next
Return MyPriceQuoteResult1
End Function
```