Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2009
August 30 - September 2, 2009, San Diego, California, USA

DETC2009-87089

SERVICE ORIENTED AND ORCHESTRATED FRAMEWORK FOR SUPPLY CHAIN INTEGRATION

Jack C.P. Cheng*
Department of Civil and Environmental Engineering
Stanford University, Stanford, California 94305, U.S.A.

Kincho H. Law
Department of Civil and Environmental Engineering
Stanford University, Stanford, California 94305, U.S.A.

Albert Jones
National Institute of Standards and Technology
Gaithersburg, MD 20899, U.S.A.

Ram Sriram
National Institute of Standards and Technology
Gaithersburg, MD 20899, U.S.A.

ABSTRACT

Supply chain management integrates key business processes and facilities, involving end users and suppliers that provide products, services, and information. Supply chain integration can potentially add value to the stakeholders along product development and manufacturing life cycle as well as the customers in terms of cost, time, service level, quality, and risk. In the manufacturing industry, there are many attempts to develop methodologies, standards, and technologies to integrate various applications for product design and manufacturing. However, studies on integrating and aligning product design and manufacturing with other operations in supply chains are relatively lacking. In an integrated supply chain, information, applications, and services are shared and become available among supply chain members within and across organizational boundaries. Existing technologies and tools such as Electronic Data Interchange (EDI) infrastructures and Enterprise Resource Planning (ERP) systems do not provide a flexible and reusable solution to information sharing and application integration. This paper presents a prototype service oriented web-based system, SC Collaborator (Supply Chain Collaborator), which leverages web services, web portal, and open source technologies to provide a flexible, customizable, and economical tool for supply chain integration. The prototype system implements a service oriented portal-based framework and allows service orchestration according to processes. This paper presents the service oriented portal-based framework, the system architecture of SC Collaborator, and the schematic representation and implementation of process models with the system. The paper also illustrates the use of the SC Collaborator system to facilitate cross-functional, cross-departmental, and cross-organizational collaborations using a bus manufacturing scenario.

1 INTRODUCTION

A supply chain consists of a network of key business processes and facilities, involving end users and suppliers that provide products, services, and information [1]. Traditionally, marketing, distribution, planning, manufacturing, and purchasing units and organizations along a supply chain operate independently. The value of integrating members along supply chains has been studied and identified in many industries [2, 3]. Supply chain integration helps reduce cost, improve responsiveness to changes, increase service level, and facilitate decision making. In an integrated supply chain, information is shared and becomes available among the members. This enhances supply chain visibility and avoids information delays and distortions. Insufficient supply chain visibility makes members vulnerable to quality and service level problems from business partners and therefore subject to risks. Information delays and distortions lead to an increase in demand signal variation along the supply chain upstream, a phenomenon called the bullwhip effect [4]. Therefore, supply chain integration is one of the keys to success for a business enterprise.

The lack of integration and interoperation of supply chains leads to significant economical costs in the manufacturing industry. According to a study by the National Institute of Standards and Technology (NIST) in 2002, imperfect interoperability costs the US automotive industry about one billion dollars per year and delays the introduction of new models by at least two months [5]. Interoperability facilitates the coordination of information flows and the integration of applications and services within and across companies participating in a supply chain. Therefore, interoperation adds value to each individual information source and application when information from multiple and likely heterogeneous

* Author of correspondence, Phone: (650) 862-3262, Email: cpcheng@stanford.edu.

sources can be accessed, related, and combined. Creating an interoperable network to support data exchange and communication among software applications can be expensive. Organizations continuously spend up to 50 percent of their information technology budgets on application integration [6]. In the manufacturing industry, there are much research efforts on the development and standardization of systems that integrate computer-aided design (CAD), computer-aided engineering (CAE), and computer-aided manufacturing (CAM) applications for product design, development, and manufacturing [7-10]. Tools that can potentially facilitate collaborations among supply chain members have also been proposed [11]. If the product development and manufacturing processes can be aligned with other business processes within and across companies, every supply chain member can potentially be benefited in terms of cost, time, quality, and value.

Integrating information and applications in a manufacturing supply chain is a non-trivial task. A manufacturing supply chain usually involves numerous participants who may use different hardware platforms and software programs. These supply chain participants may also use different representations and perspectives to describe their information. Some companies may even develop their own software programs and information models in-house and require their trading partners to support these programs and models for data exchange. Currently, integration and interoperation of these software applications and information are commonly performed through application-to-application interfaces on a case-by-case basis. These ad-hoc integration infrastructures often are not reusable and transportable to another project. The use of application-to-application interfaces leads to the N-square problem and is not scalable for integration in the scope of entire supply chains. Therefore, we need a methodology to integrate information, applications, and services in a flexible, scalable, and reusable manner.

With rapid development of communication technology, the Internet has become ubiquitous and instantaneously accessible. The proliferation of the Internet makes it the most cost effective means of driving supply chain integration and information sharing [12]. Companies increasingly take advantage of the Internet to create a virtual value chain where individuals and business partners can communicate and collaborate with each other. Types of Internet-based integration and collaborations include B2C (business-to-consumer), B2B (business-to-business), G2B (government-to-business), and B2E (business-to-employee).

Utilizing the Internet as the communication network, the web services technology has emerged as a promising tool to integrate distributed information sources and software functionalities in a flexible, scalable, and reusable manner. Web services technology enables a service oriented approach of integration of information and applications, which are deployed into individual web service units and distributed on a network which supply chain members can connect to. The service units act as the components for various business functions and can be aggregated into a complex business process or workflow. The service units can also be reused, avoiding repeated development of the same functionality.

Web portals have been used for information repository and sharing within a company as well as across organizations. The customizability of layouts and access control in a web portal provides a secure, tailored way to deliver the right information to the right user at the right time. Functionalities of current web portals are not flexible and extendable. In a service oriented portal-based framework, however, information, application functionalities, and system operations are deployed as discrete web service units, which can be flexibly combined according to different needs. Exposing the web service units also allows external systems and applications to retrieve information and to integrate the operations from the service oriented system, increasing the usability of the system, and enabling automation of business processes.

The paper is organized as follows: Section 2 discusses some of the shortcomings of current methods and tools for supply chain integration, and how they can be avoided using a service oriented approach. Section 2 also describes the service oriented portal-based framework, a means to integrate information, applications, and services. Section 3 presents the system architecture of a prototype web-based system, namely SC Collaborator (Supply Chain Collaborator), that we developed based on the service oriented portal-based framework. Process models can be incorporated in the SC Collaborator system to orchestrate discrete web service units. Section 3 also describes the schematic representation and a practical implementation of the process models. Section 4 presents an example scenario to demonstrate the potential of SC Collaborator to integrate services and facilitate collaborations in a cross-functional, cross-departmental, and cross-enterprise supply chain.

2 SERVICE ORIENTED APPROACH FOR INTEGRATION

2.1 Current Practices for Supply Chain Integration

Information, applications, and services need to be integrated in a supply chain to facilitate cross-functional, cross-departmental, and cross-enterprise collaborations among individuals and companies. Some companies establish communication networks using standards such as Electronic Data Interchange (EDI) to connect and exchange data with partners [13-15]. However, the implementation of such communication infrastructures usually requires high cost and long configuration time, partly due to the lack of information standardization among trading partners. Small and medium-sized businesses may be reluctant to afford a large investment for the infrastructures. The long configuration time reduces the flexibility of product introduction, development, and production.

Some companies have adopted various supply chain management tools such as enterprise resource planning (ERP) systems to integrate loosely distributed information and applications within and across companies in the manufacturing industry [16-18]. An ERP system is typically employed to seamlessly integrate all the information flowing through the company such as finances, accounting, human resources, supply chain, and customer information [19]. ERP systems can potentially enhance transparency across the supply chain by eliminating information distortions and increase

information velocity by reducing information delays [20]. Many corporations have implemented ERP systems to facilitate their front-end customer relationship and to support their back-end operations.

Adoption of ERP systems, however, does not often result in significant improvement in project performance as expected. One study estimated that 96.4% of ERP implementations failed [21] whereas another study reported that 70% of ERP implementations did not achieve their estimated benefits [22]. ERP systems have many limitations such as implementation complexity, integration problems, and customization problems [23]. Akkermans et al. [20] conducted an exploratory study on commercial implementation of ERP systems and concluded four major limitations of ERP systems that often led to unexpected underperformance of these tools: (1) inflexibility to accommodate changes of supply chain structures, (2) lack of modular and open system architecture, (3) lack of functionality beyond managing transactions, and (4) inability to share internal data efficiently with supply chain partners across organizational boundaries. We believe that a system based on the service oriented architecture (SOA) can address many of these major limitations.

2.2 Service Oriented Architecture (SOA)

SOA is a software development paradigm in which information and applications are deployed and distributed as individual service units, which can be invoked over a network and combined to solve complex business problems. There are many benefits for adopting SOA in information systems. The "plug-and-play" capability of these service units allows convenient and flexible reconfiguration of system functionalities. Furthermore, the system can be divided into modules for development and maintenance, allowing a large complicated system to be built in a scalable manner. In addition, by invoking various applications via the service units, a service oriented system can extend its functionalities for sophisticated business tasks. Last but not least, internal data can be shared easily and efficiently among organizations in SOA as the data are delivered as service units that can be accessed on a standardized protocol.

Web services are the building blocks of SOA. A web service can be described as a specific function that is distributed on the Internet to provide information or services to users through standardized application-to-application interactions. Leveraging well established Internet protocols and commonly used machine readable representations, web services can be located, invoked, combined, and reused. Web services can create dynamic responses and are different from conventional websites, which deliver only static information. Web services are self-contained and self-describing. They are also encapsulated, meaning that integrated web services can be updated or replaced without affecting the functionality or integrity of other independent services. Web services technology enables interoperation of applications because programs written in different languages and operating on different systems can be integrated via standardized protocols.

2.3 Service Oriented Portal-based Framework

Loosely coupled web services can be aggregated using web portal technology. A web portal is a web-based system that acts as a gateway to a larger system or a network of web applications. It is a useful tool to aggregate scattered, distributed information and services into a single point of access regardless of their location or storage mechanism. The basic operational units of a portal system are web portlets, which are sub-programs that encapsulate a single or a number of web applications. Through the portal system, multiple information sources and applications can be accessed, retrieved, and integrated into a workflow or a supply chain.

Web portals are commonly used to build an intranet for content and document management within organizations [24]. They serve as a repository of information and documents for data storage, publication, and retrieval. Due to their security and customizability, web portals allow users to securely access sensitive personal information, and enable system administrators to manage a huge amount of information in a centralized manner. There is also a trend to build portal systems for cross-organizational collaboration. However, there is little, if any, rigorous research on portal design, development, maintenance, and updating for facilitating supply chain management decisions [25]. Little effort has been made to use web portal technology for collaboration and sharing of information and functionality to support decision making among multiple partners.

For the service oriented portal framework developed in this study, system operations, applications, and information sources are wrapped and deployed as individual web services, which can be located and invoked by application portlet units. As shown in Figure 1, every application portlet unit can integrate a single or multiple system operations, applications, or information web services into a workflow for specific business processes. These web services can be reused by different portlet units in different workflows, or used by a single portlet unit multiple times in one workflow. As a result, the development of repeated system operations is avoided, and applications and information sources can be used concurrently. In addition, modification of system functionalities becomes easy and quick as every business process is divided into separate reusable web service components. Communications among application portlet units are allowed, supporting aggregation and interaction of service workflows.

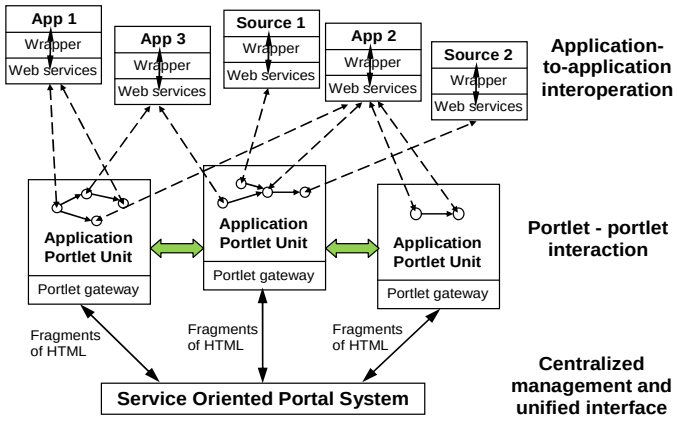

Figure 1. Conceptual framework for service oriented portal system.

3 SC COLLABORATOR FRAMEWORK

SC Collaborator (Supply Chain Collaborator) is a prototype system developed using portal and web services technologies. Based on a service oriented portal framework, SC Collaborator implements the SOA approach and provides modularity, flexibility, extendable functionality, customizability, and single point of access. The system supports Single Sign-On (SSO), enabling users to gain access to multiple applications in the system with only one login. Open source technologies are leveraged to minimize implementation costs which hinder the usability in small and medium-sized enterprises. Specifically, open source software Apache Tomcat [26], Liferay Portal [27], and MySQL [28] are utilized to support the communication channels, user interface, and database support of the SC Collaborator system. Figure 2 shows the homepage of the prototype system for users to log in.

Figure 2. Homepage of SC Collaborator system.

3.1 System Architecture

Figure 3 shows the system architecture of the SC Collaborator framework. The framework consists of an access control engine, a database support, and four layers of integrated functionalities – a communication layer, a portal interface layer, a business applications layer, and an extensible computing layer. The communication layer provides a communication channel for users to access the system. The portal interface layer serves as a unified and customizable platform to support interactions between users and the system. The business applications layer provides a framework for executing various business processes such as knowledge management and decision making. The extensible computing layer consists of numerous databases, software applications, and web services that the business applications layer can integrate to support high-level or computationally intensive business functions. In the following, we will discuss the three layers and the database support component of SC Collaborator in detail.

3.1.1 Communication layer

A user-friendly and readily accessible communication channel is essential to the usability of a system. The SC Collaborator system uses an open source platform – Apache Tomcat [26] – to enable the connectivity and access to the system. Apache Tomcat serves as a web servlet container for the communication servlets, Struts [29] and Axis [30]. While some information systems require the client-side to install particular communication software in order to be connected, the Struts servlet that resides in SC Collaborator allows users to access the system using web browsers, which are commonly available on every computer. The Struts servlet also enables remote users to access the system using wireless devices such as cell phones and personal digital assistants (PDA) via the Wireless Application Protocol (WAP).

The Axis servlet that resides on the communication layer enables system operations of the SC Collaborator system to be exposed as standard web services. The deployed web services are described in standardized Web Service Definition Language (WSDL) [31] files for service discovery, description, and invocation. Figure 4 shows the WSDL file of a simple system operation which sends purchase orders to suppliers. Users can request information from the system and execute internal operations via the Simple Object Access Protocol (SOAP) [32].

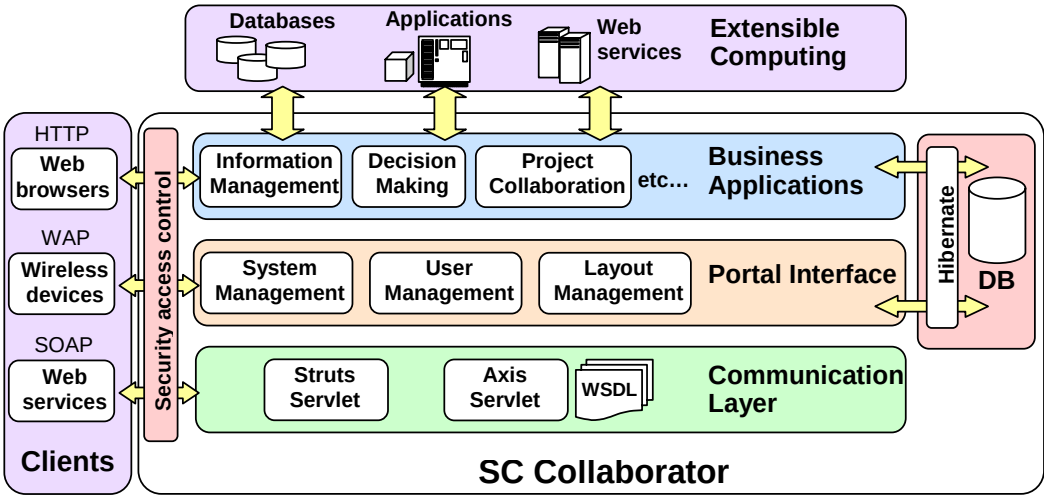

Figure 3. System architecture of SC Collaborator system.

Figure 4. Example WSDL deployed in SC Collaborator.

3.1.2 Portal interface layer

Web portal technology is leveraged to provide a flexible and customizable user interface in the system. The portal user interface of the SC Collaborator system is managed in separate modules. Every module represents a project, an organization, or a group of similar business functionalities. A single module contains a number of submodules, each of which can integrate multiple application portlet units. Configuration, permissions, and layout can be configured for each module, submodule, and portlet. System management also includes activity logging, user tracking, computer resources utilization, and connection limits setting. It helps the system administrators evaluate the system and configure it to suit different needs.

User management can be performed at the levels of individual users, organizations, user groups, and roles. A user is an individual who performs tasks in the system. An organization represents a corporate hierarchy. A user group is a grouping of users. Unlike organizations, user groups have no context associated with them. They are grouped to facilitate communications and aid system administrators to assign permissions and roles to a group of users. A user can be associated with any number of user groups, but only one organization. Every member inherits the roles and permissions that are assigned to the organization or user group that the member belongs to. SC Collaborator is a role-based system. A role is a collection of permissions. The types of roles in the SC Collaborator system are system administrator, module administrator, advanced user, module member, normal user, and guest. Each role has its predefined set of permissions to the system, layout, modules, submodules, and portlets.

The user interface for web browsers and wireless devices can be configured through the layout management portlet unit. The portlet unit allows users with either a system administrator role or a module administrator role to add and delete submodules, to set up the permissions of submodules, and to configure the submodule style. On each submodule, the administrative users can add, delete, and allocate application portlet units. The administrative users can also grant individual users the permissions to view, modify, and configure a specific module, submodule, and portlet. Therefore the system layout can be highly customizable so that some modules or portlet units are available only to the designated users, organizations, or user groups. This ensures that the right information is delivered to the right person at the right time.

3.1.3 Business applications layer

Each application portlet unit is an independent unit, which performs a specific task or business process. Based on the Java framework, a portlet unit can perform computations, execute other applications, connect to databases, and invoke web services. Therefore, multiple services can be integrated in a single portlet unit to implement various business processes. For instance, the application portlet unit that helps retailers manage the purchase orders they have submitted integrates three different services: (1) service that submits purchase orders to manufacturers, (2) service that monitors the status of each purchase order, and (3) service that triggers warning notifications when a problem is encountered. A portlet unit in SC Collaborator can also interact with other portlets to solve complicated business problems. The application portlet units in SC Collaborator are compliant with Java Specification Request (JSR) 168 standard [33], a specification that defines a standard programming model for portlet development. As a result, the portlet units can be packaged and reused by other portal systems, allowing high portability across platforms.

3.1.4 Database support

In the database tier, an open source database – MySQL – is used to store the application data as well as the system information including user information, layout configurations, and user and system settings. The SC Collaborator system is not bounded to a particular database system. The system can be installed with any Java Database Connectivity (JDBC) [34] compliant database without any complicated configuration and modification of codes due to the use of the Hibernate framework [35]. The Hibernate framework maps the objects in a relational database into object-oriented Java classes. If a user has already installed other databases such as PostgreSQL and Oracle database, SC Collaborator can integrate with the existing database with little effort. The user does not need to install and execute MySQL in order for SC Collaborator to run.

3.2 Process Models for Service Orchestration

The SC Collaborator system is a service oriented portal-based framework where information, applications, and internal system operations are deployed and delivered as web services. These internal web services are aggregated with external web services on the business applications layer into workflows to perform complex tasks. To implement a business process "add purchase order," for example, cross-application activities may be required, such as adding a purchase order to the production plan, sending confirmation to the customer, changing the status of the order and the corresponding items, and allocating materials and resources to fulfill the order. Each of these task components could be separated and deployed as individual web services. A mechanism to combine these task component services is necessary to complete a business process. There are several research efforts on web-based service integration [36, 37], which study the mechanisms to invoke, terminate, and combine web services. In SC Collaborator, the composition and orchestration of web services are performed and managed by business process models residing on the applications layer.

Figure 5 shows the schema of a service unit in a process model. Each service unit consists of six components: (1) input information, (2) output information, (3) organization involvements, (4) resources, (5) internal knowledge, and (6) implementation logic. Input information describes the types and data models of the information that is provided either by automatic event trigger programs or by manual inputs through

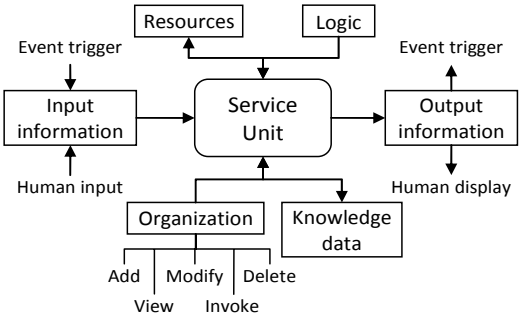

Figure 5. Components of a service unit in a process model.

tools such as web browsers. Output information describes the types and data models of the information that the service unit delivers. Organization involvements define the access rights which are implemented in the security access engine. Resources are the application functionalities or operations necessary for the service implementation. Internal knowledge is the collection of data, either hard-coded in the service unit or available from databases and data files, which supplements the input information to perform the service. Implementation logic specifies the conditions and reasoning in the service unit.

After defining the service units, they need to be articulated in a process flow model for every business process. There are

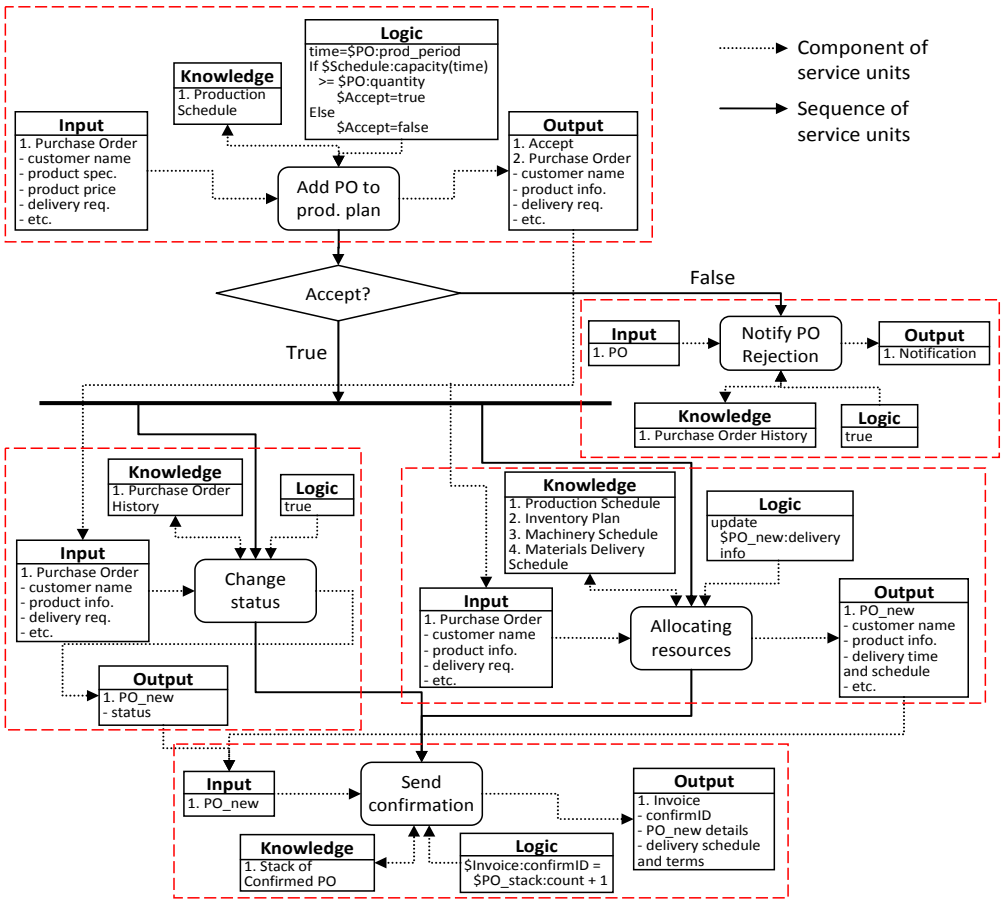

Figure 6. Process flow model for the business process "add purchase order" with schematic definitions of every service unit.

many research studies on workflow modeling and management [38-41]. In the SC Collaborator system, a process flow model is a representation that defines the interactions among various service units. The model describes the sequence of service units and the logic involved during the transitions between the services units. Every business task in SC Collaborator is associated with a single process flow model. Therefore, one application portlet unit in SC Collaborator may contain multiple process flow models, depending on the number of business tasks the portlet unit performs. For example, the portlet unit responsible for managing received purchase orders contains three process flow models for the three business tasks – receiving purchase orders, monitoring the status of each order, and notifying corresponding members when an order is updated. Figure 6 depicts the process flow model for the business process that receives and processes purchase orders.

For implementation, the process flow model is converted into Business Process Execution Language (BPEL) [42], an emerging service orchestration standard for describing the behavior of web services at different levels of abstraction. BPEL is a layer on top of WSDL and XML Schema, with WSDL and XML Schema defining the structural aspects of service interactions, and BPEL defining the behavioral aspects. BPEL supports two kinds of activity coordination – primitive activities and structured activities. Primitive activities correspond to atomic actions such as message exchange and service initiation that are being performed within a process. Examples of primitive activities include invoke, receive, reply, assign, and terminate. Structured activities impose behavioral and execution constructs on a set of activities contained within them. Structured activities can be nested and combined in arbitrary ways, thus enabling the

Figure 7. Excerpt of the BPEL file for the business process "add purchase order."

presentation of complex structures. Examples of structured activities are sequence, switch, while, and flow. Figure 7 shows an excerpt of the BPEL file converted from the process flow model for the business process "add purchase order."

4 SCENARIO EXAMPLE

To demonstrate the potential of SC Collaborator for integrating manufacturing supply chains, the system has been tested in a bus manufacturing scenario. As illustrated in Figures 8 and 9, the scenario involves cross-functional, cross-departmental, and cross-organizational interactions and collaborations. In Figure 8, a circle in an operation means that proceeding to the next operation requires conditional reasoning.

In the scenario, the manufacturer is divided into five functional departments (Figure 9). The sales department of the bus manufacturer receives purchase orders from customers. If the order contains special requests, the sales department may need to pre-process the order and invite specialized experts to the order evaluation process. The order evaluation process

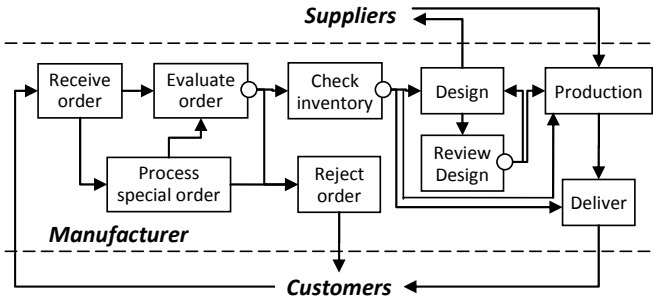

Figure 8. Operations involved in the example scenario.

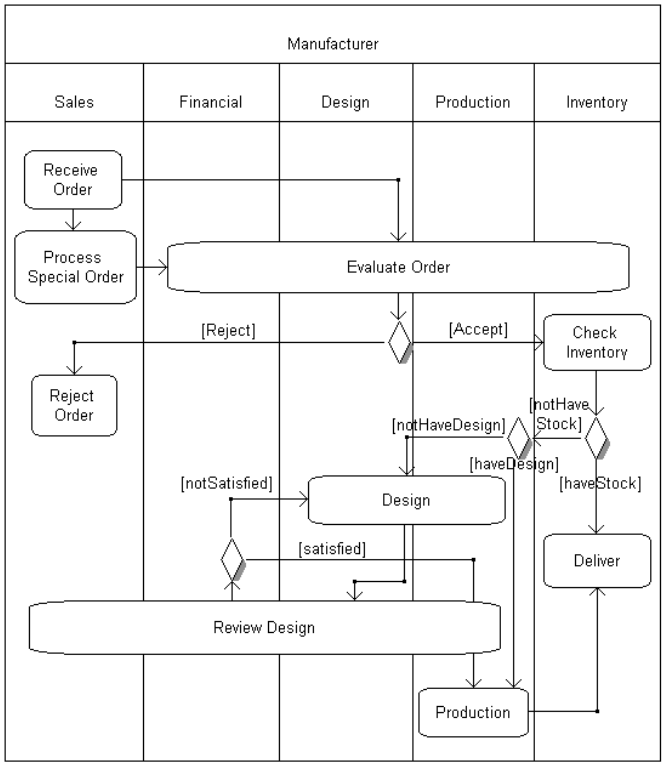

Figure 9. Involvements of the five departments of the bus manufacturer in the scenario.

involves the financial department, the inventory department, the design department, and the production department. If the order is not accepted, the sales department notifies the customer of the rejected order; otherwise, the inventory department checks the inventory and delivers the products if they are ready to delivery. If there is no finished product that satisfies the order and a new design is needed, the design department designs the products with aid from the production department. The production department may start contacting suppliers and preparing procurement at the same time. The design is then available for review, which is a recurring process among various departments. If the design has been done previously, the design department can reuse the existing design and the production department begins the procurement process. When the design is ready, the production department begins the production process and then sends the finished products to the inventory department for delivery.

The system is customized so that members are provided with different information and functionalities, depending on the organizations and departments the members belong to. For example, the system layout for members in the sales department contains submodules for monitoring and processing orders, inviting specialized experts to evaluate orders, and communicating with customers, which are not included in the layout for members in the design department when the system is logged in. The system can also be personalized so that each individual is granted with different access rights according to the user profiles. For instance, the system allows junior members in the design department to view order evaluations, to submit product designs, and to retrieve design reviews while senior members are also allowed to participate in the discussions in the order evaluation and the design review processes.

Take the "receive order" and "evaluate order" processes as an example. All the five departments of the bus manufacturer participate in managing and evaluating received orders. As shown in Figure 10, some service units in the system are specific to particular departments and require human inputs, whereas some are executed on the system level automatically. When the SC Collaborator system of the bus manufacturer receives a purchase order, it sends a notification message to the sales department and allows the department to view the order. The sales department validates the order and determines the need for additional people in the order evaluation process (Figure 11). If the order is valid, the system invokes multiple order evaluation service units. The order is evaluated by various departments in the aspect of customer, product configuration, and product delivery. For instance, the design department gathers information from the sales department and the financial department, evaluates the product configuration requested in the order, tries alternative configuration designs, and provides cost estimate and comments using the SC Collaborator system (Figure 12). The sales department can then refer to the evaluations from various departments and decide to accept or reject the order using the system layout shown in Figure 11. After that, other service units are activated if the order is accepted. The service oriented architecture of SC Collaborator enables quick reconfiguration of the operation workflow and flexible changes of organizational structures and involvements. It

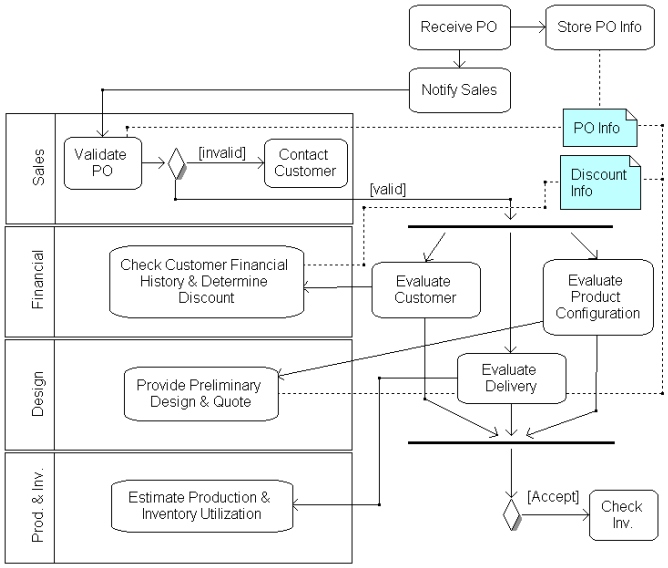

Figure 10. Service units involved in the "receive order" and "evaluate order" processes.

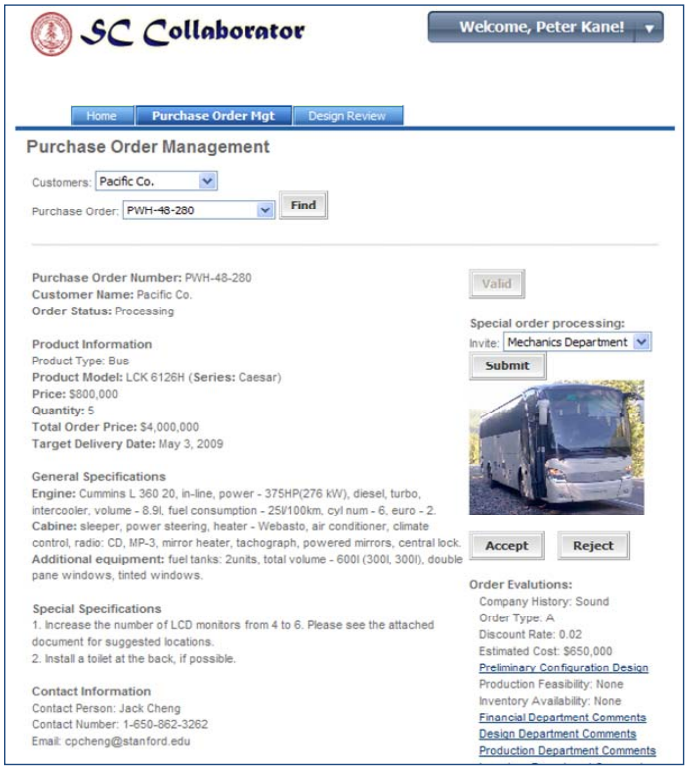

Figure 11. System layout when a member in the sales department logs in to check the status of order evaluation.

eases the modification and deployment of the system when a new sales channel is added, the suppliers are changed, or a new product series is introduced.

5 CONCLUDING REMARKS

The benefits of supply chain integration are well recognized in various industries. Information is often delayed, missing, distorted, or incomplete in a supply chain. Sharing information and integrating applications and services among supply chain

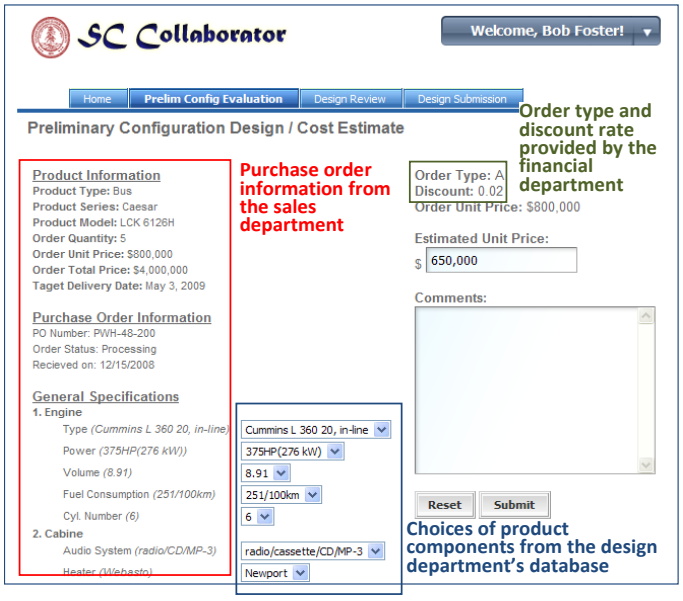

Figure 12. Screenshot of the SC Collaborator system showing a senior member in the design department performing preliminary product component configuration design and cost estimating.

members can therefore help reduce cost, increase responsiveness and service level, support decision making, and enhance project transparency. There are numerous commercial tools on the market for managing and integrating supply chains. However, these tools are not flexible, scalable, and extensible. The service oriented approach provides an alternative solution to connect individuals and to aggregate scattered information and applications along supply chains.

This paper presents a service oriented portal-based framework. Leveraging web portal technology, the framework offers customizable system layout and access control. With service oriented approach, the framework enables modular development and flexible reconfiguration of system functionalities. Internal information, application functionalities, and system operations are separated and deployed as individual service units, which can be located, invoked, and retrieved via standardized protocol. These service units can be reused and combined to create a complex business process. Interoperability problem among applications is also addressed because programs written in different languages and operating on different systems can be integrated via standardized protocols.

Based on the service oriented portal-based framework, we have utilized web services, web portal, and open source technologies to develop a prototype web-based system, namely SC Collaborator, for supply chain integration. The system is comprised of an access control engine, a database support, and four layers of integrated functionalities. The combination and orchestration of service units are performed on the business applications layer with the aids of process models that reside in individual application portlets. Each service unit in the process model contains six components – input information, output information, organization involvements, resources, internal knowledge, and implementation logic. The schema and implementation of the process models are illustrated in this paper. An example bus manufacturing scenario is also included to demonstrate the use of SC Collaborator to facilitate integration within and among organizations.

6 ACKNOWLEDGEMENT

The authors would like to acknowledge the supports by the US National Science Foundation (NSF), Grant No. CMS-0601167, the Center for Integrated Facility Engineering (CIFE) at Stanford University, and the Enterprise Systems Group at the National Institute of Standards and Technology (NIST). The first two authors at Stanford University would like to thank Prof. Jian Cao of Shanghai Jiao Tong University for the discussions during his visit to Stanford and the suggestions for the bus manufacturing example scenario presented in Section 4. Any opinions and findings are those of the authors, and do not necessarily reflect the views of NSF, CIFE, or NIST. No approval or endorsement of any commercial product by NIST, NSF, or Stanford University is intended or implied.

7 REFERENCE

[1] Lambert, D. M., Cooper, M. C. and Pagh, J. D., 1998, "Supply Chain Management: Implementation Issues and Research Opportunities," *The International Journal of Logistics Management*, Vol. **9** (2), pp. 1-19.

[2] Morash, E. A. and Clinton, S. R., 1998, "Supply Chain Integration: Customer Value through Collaborative Closeness versus Operational Excellence," *Journal of Marketing Theory and Practice*, Vol. **6** (4), pp. 104-20.

[3] Simatupang, T. M., Wright, A. C. and Sridharan, R., 2002, "The Knowledge of Coordination for Supply Chain Integration," *Business Process Management Journal*, Vol. **8** (3), pp. 289-308.

[4] Lee, H. L., Padmanabhan, V. and Whang, S., 2004, "Information Distortion in a Supply Chain: The Bullwhip Effect," *Management Science*, Vol. **50** (12 Supplement), pp. 1875-1886.

[5] Brunnermeier, S. B. and Martin, S. A., 2002, "Interoperability Costs in the US Automotive Supply Chain," *Supply Chain Management: An International Journal*, Vol. **7** (2), pp. 71-82.

[6] Bingi, P., Sharma, M. K. and Godla, J. K., 2001, "Critical Issues Affecting an ERP Implementation," *Enterprise Systems Integration*, Auerbach Publications, CPC Press.

[7] Deng, Y. M., Britton, G. A., Lam, Y. C., Tor, S. B. and Ma, Y. S., 2002, "Feature-Based CAD-CAE Integration Model for Injection-Moulded Product Design," *International Journal of Production Research*, Vol. **40** (15), pp. 3737-3750.

[8] Deng, Y. M., Lam, Y. C., Tor, S. B. and Britton, G. A., 2002, "A CAD-CAE Integrated Injection Molding Design System," *Engineering with Computers*, Vol. **18** (1), pp. 80-92.

[9] Lee, H. S. and Chang, S. L., 2003, "Development of a CAD/CAE/CAM System for a Robot Manipulator,"

Journal of Materials Processing Technology, Vol. **140** (1-3), pp. 100-104.

[10] Lee, S. H., 2005, "A CAD-CAE Integration Approach Using Feature-Based Multi-Resolution and Multi-Abstraction Modelling Techniques," *Computer-Aided Design*, Vol. **37** (9), pp. 941-955.

[11] Sriram, R. D., 2002, *Distributed and integrated collaborative engineering design*, Sarven Publishers.

[12] Lee, H. L. and Whang, S., 2005, "Supply Chain Integration over the Internet," *Supply Chain Management: Models, Applications, and Research Directions*, Springer US.

[13] Emmelhainz, M. A., 1989, *Electronic Data Interchange: A Total Management Guide*, Van Nostrand Reinhold Co., New York, NY, USA.

[14] Hill, C. A. and Scudder, G. D., 2002, "The Use of Electronic Data Interchange for Supply Chain Coordination in the Food Industry," *Journal of Operations Management*, Vol. **20** (4), pp. 375-387.

[15] Kaefer, F. and Bendoly, E., 2000, "The Adoption of Electronic Data Interchange: A Model and Practical Tool for Managers," *Decision Support Systems*, Vol. **30** (1), pp. 23-32.

[16] Muscatello, J. R., 2003, "Implementing Enterprise Resource Planning (ERP) Systems in Small and Midsize Manufacturing Firms," *International Journal of Operations and Production Management*, Vol. **23** (7/8), pp. 850-871.

[17] Duplaga, E. A. and Astani, M., 2003, "Implementing ERP in Manufacturing," *Information Systems Management*, Vol. **20** (3), pp. 68-75.

[18] Koh, S. C. L. and Saad, S., 2004, "Modeling Uncertainty Under a Multi-Echelon ERP-Controlled Manufacturing System," *Journal of Manufacturing Technology Management*, Vol. **15** (3), pp. 239-253.

[19] Davenport, T. H., 1998, "Putting the Enterprise into the Enterprise System," *Harvard Business Review*, Vol. **76** (4), pp. 121.

[20] Akkermans, H. A., Bogerd, P., Yücesan, E. and van Wassenhove, L. N., 2003, "The Impact of ERP on Supply Chain Management: Exploratory Findings from a European Delphi Study," *European Journal of Operational Research*, Vol. **146** (2), pp. 284-301.

[21] Rao, S. S., 2000, "Enterprise Resource Planning: Business Needs and Technologies," *Industrial Management and Data Systems*, Vol. **100** (1), pp. 81-8.

[22] Al-Mashari, M., 2000, "Constructs Of Process Change Management in ERPContent: A Focus on SAP R/3," *Proceedings of 2000 Americas Conference on Information Systems, AMCIS 2000*, Long Island, California, USA, pp. 977-980.

[23] Themistocleous, M., Irani, Z., O'Keefe, R. M. and Paul, R., 2001, "ERP Problems and Application Integration Issues: An Empirical Survey," *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, Hawaii, USA.

[24] Michelinakis, D., 2004, "Open Source Content Management Systems: An Argumentative Approach," Ph.D. thesis, University of Warwick.

[25] Vakharia, A. J., 2002, "e-Business and Supply Chain Management," *Decision Sciences*, Vol. **33** (4), pp. 495-504.

[26] Apache, 2007, Apache Tomcat 5.5.

[27] Liferay Inc., 2008, Liferay Open Source Enterprise Portal System.

[28] Sun Microsystems, 2007, MySQL 5.0.

[29] Apache, 2008, Apache Struts.

[30] Apache, 2006, Apache Axis.

[31] Booth, D. and Liu, C. K., 2004, "Web Services Description Language (WSDL) Version 2.0 Part 0: Primer," *W3C Working Draft 21*.

[32] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H. F., Thatte, S. and Winer, D., 2000, Simple Object Access Protocol (SOAP) 1.1.

[33] Abdelnur, A., Chien, E. and Hepper, S., 2003, "JSR 168: Portlet Specification," *Java Specification Requests, Java Community Process, Sun Microsystems and IBM*.

[34] Sun Microsystems, 2002, JDBC Data Access API.

[35] Red Hat, 2008, Hibernate framework.

[36] Greenwood, D., Calisti, M., Ag, W. T. and Zurich, S., 2004, "Engineering Web Service-Agent Integration," *Proceedings of 2004 IEEE International Conference on Systems, Man and Cybernetics*, pp. 1918 - 1925.

[37] Cheng, J., 2004, "A Simulation Access Language and Framework with Applications to Project Management," Ph.D. thesis, Stanford University, Stanford, CA.

[38] Chiu, D. K. W., Li, Q. and Karlapalem, K., 1999, "A Meta Modeling Approach to Workflow Management Systems Supporting Exception Handling," *Information Systems*, Vol. **24** (2), pp. 159-184.

[39] van der Aalst, W., 2000, "Loosely Coupled Interorganizational Workflows: Modeling and Analyzing Workflows Crossing Organizational Boundaries," *Information & Management*, Vol. **37** (2), pp. 67-75.

[40] Yu, J. and Buyya, R., 2005, "A Taxonomy of Workflow Management Systems for Grid Computing," *Journal of Grid Computing*, Vol. **3** (3), pp. 171-200.

[41] Zur Muehlen, M., 2004, "Organizational Management in Workflow Applications–Issues and Perspectives," *Information Technology and Management*, Vol. **5** (3), pp. 271-291.

[42] BEA Systems, Microsoft, IBM and SAP, 2003, Business Process Execution Language for Web Services (BPEL4WS).