A Distributed Portal-Based Platform for Construction Supply Chain Interoperability

C.P. Cheng & K.H. Law
Engineering Informatics Group, Stanford University, CA 94305, USA

H. Bjornsson
Chalmers University of Technology, Gothenburg, Sweden

ABSTRACT: Collaboration and interoperability plays a very significant role in a construction supply chain. The lack of information sharing and software interoperability hampers coordination and collaboration among project participants, which are crucial to the success of a supply chain. As the Internet becomes ubiquitous and instantaneously accessible, the web services model has emerged as a promising approach to support supply chain collaboration and achieve interoperability. This paper discusses a web service framework to connect, invoke and integrate loosely coupled, heterogeneous information sources and software applications. Specifically, web portal technology is leveraged to implement the web services platform and to provide a robust and customizable user interface. The issues of information ownership rights and proprietary privacy may hinder information sharing among companies in a centralized portal system. A decentralized portal network architecture is introduced to promote sharing of information and to enable distributed computing. This paper presents the prototype web-based platform, *SC Collaborator*, that is designed to support data exchange, information sharing and supply chain collaboration. The platform leverages open source technologies and implements the proposed distributed portal network architecture. Two example scenarios are included to demonstrate the potential of the SC Collaborator system in managing information flows in AEC activities and facilitating interoperability.

1 INTRODUCTION

Construction supply chains are highly fragmented in nature. The scattered information sources and hardware platforms pose a challenge to the establishment of data exchange and communication channels in the architectural, engineering and construction (AEC) industry (Lee & Bernold 2008). Integrating information and software functionality in a multi-participant supply chain is a non-trivial task because project participants may use different hardware platforms and software applications. To facilitate the coordination of information flows within and across companies participating in a supply chain, the interoperability issue among distributed hardware platforms, software applications and information sources must be addressed. The purpose of interoperation is to increase the value of information when information from multiple and, likely, heterogeneous sources is accessed, related and combined. Interoperability allows data exchange and sharing among heterogeneous information sources, software applications and systems. Therefore, interoperation can add significant value to each individual source and application, as well as enhances efficiency and performance of a supply chain.

The lack of interoperability leads to significant economical costs in various industries. According to a study by NIST, in the year 2002 alone, inefficient interoperability costs more than $15.8 billion to the construction industry on the design, construction and maintenance of large commercial, institutional and industrial buildings (Gallaher et al. 2004). Another study by NIST estimates that imperfect interoperability costs the US automotive industry about one billion dollars per year (Brunnermeier & Martin 2002). Creating interoperable networks to support data exchange and communication among software applications can be expensive. As reported in Bingi et al. (2001), organizations continuously spend up to 50 percent of their IT budgets on application integration.

Insufficient interoperability also hinders project efficiency and coordination. Lack of information sharing and application integration results in the myopic control in a supply chain. Project participants work in an isolated manner and do not have a macroscopic view of the whole supply chain. The participants make their own decisions to optimize their costs or benefits, based on the information from the immediate downstream node in the supply chain. As a result, the whole system may not achieve the

Figure 1. SC Collaborator viewed in web browsers.

optimum performance even though each project participant optimizes its own performance (Yu et al. 2001). This myopic view may also lead to information distortion among supply chain partners. In the procurement process, for example, as individuals attempt to forecast and make decisions based on the information from the immediate downstream node only, demand signal amplification and variation tend to increase as one moves upstream – a phenomenon termed “bullwhip effect” (Lee et al. 2004). Therefore, sharing of information and collaboration between each party is crucial to the success of a supply chain (Gunasekaran & Ngai 2004, Lee et al. 2000). Supply chain interoperability enhances the performance of construction projects in terms of quality, time, cost and value.

The issue of interoperability can be tackled on two levels, namely, information interoperability and software interoperability. To facilitate data sharing and information interoperability, much efforts have been spent to build unifying standard models of concepts and definitions such as Industry Foundation Classes (International Alliance for Interoperability 1997), CIMSteel Integration Standards (Watson & Crowley 1995) and BIM (Brucker & Nachtigall 2005) to capture various phases and facets of design and construction processes. In practice, software services that need to communicate will likely be based on distinct ontology or data models and structures, that reflect the contexts and vocabularies for specific applications and domains (Ray 2002). Efforts have been attempted to “harmonize” different information ontological standards (Begley et al. 2005, Cheng et al. 2008, Cheng et al. 2002, Lipman 2006). This paper focuses on software interoperability to facilitate the connectivity, deployment and integration of distributed information sources and engineering services.

In this paper, we will first review the current approaches to integrate information, applications and services into workflows and supply chains. The proposed service integration framework which leverages web services model and web portal technologies will then be discussed. We will present a prototype platform, *SC Collaborator* (Fig. 1), that is designed to support interoperation and collaboration in AEC supply chains. Two example scenarios will be included to demonstrate the potential use of the prototype system for the management of information flow in AEC supply chains.

2 INTEGRATION OF INFORMATION, APPLICATIONS AND SERVICES

Information and software applications can be integrated in a local machine relatively easily. Sharing and exchange of information can be performed across applications as long as application programming interface (API) is available for the mapping applications. Invocation and integration of software functionality are also allowed for some applications through their APIs. In a supply chain, however, companies and project participants are geographically distributed and employ different hardware platforms and software applications. Mechanisms which support the integration of information and applications in a local machine do not facilitate interoperability in a supply chain. Some companies establish communication networks using standards such as Electronic Data Interchange (EDI) to connect with partners. However, the implementation of such communication network infrastructures can be very expensive and is not economically feasible for small to medium businesses, which are common in the construction industry. In addition, the long configuration time and rigid system architecture do not provide the flexibility needed to support fast changing construction supply chain.

With the rapid development of web technologies, 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 (Lee & Whang 2005). Companies increasingly take advantage of the Internet and information technology to create a virtual e-chain to communicate and collaborate with other supply chain participants (Manthou et al. 2004). Various efforts have been made to leverage the Internet for information and service integration, engineering simulation, negotiation and cooperation, collaborative planning and design, and other supporting activities (Anumba & Duke 1997, Cheng et al. 2001, Cheng et al. 2006, Danso-Amoako et al. 2006).

The web services model has emerged as a promising approach to connect and aggregate distributed web applications and information sources. Utilizing the Internet as the communication network, a “web service” can be described as a specific function that is delivered over the web to provide information of services to users. Information sources and software applications can be packaged into individual web service components. Leveraging well established Internet protocols and commonly used machine readable representations, web services can be located, invoked, combined and reused. The implementations of web services are encapsulated and not exposed to the users. Changing the implementation of one web service function does not alter the way that the users invoke the function. This enables clean and robust deployment and maintenance of web services. With the service oriented architecture, the web services model allows a large complicated system infrastructure to be built in a scalable manner. Modular system development and maintenance is enabled as the system is divided into web service components which can be managed separately. The web service components can be plug-and-played to cater different project requirements at each project stage. The reusability of the components also reduces the time and efforts to develop similar components.

There are many existing mechanisms to deploy, invoke, orchestrate and terminate web services for web-based integration (Cheng 2004, Danso-Amoako et al. 2006, Greenwood et al. 2004, Ismail et al. 2002). Various languages have been proposed to facilitate the discovery, execution and composition of web services. Examples include Web Services Description Language (WSDL) (Booth & Liu 2004) and Flow Language (WSFL) (Leymann 2001), Business Process Execution Language for Web Services (BPEL4WS) (Andrews et al. 2003), and Web Service Ontology based on DARPA Agent Markup Language (DAML-S) (Ankolekar et al. 2001). Semantic web services have also been an active area of research and development (de Bruijn et al. 2005, Preist 2004).

3 DISTRIBUTED PORTAL NETWORK

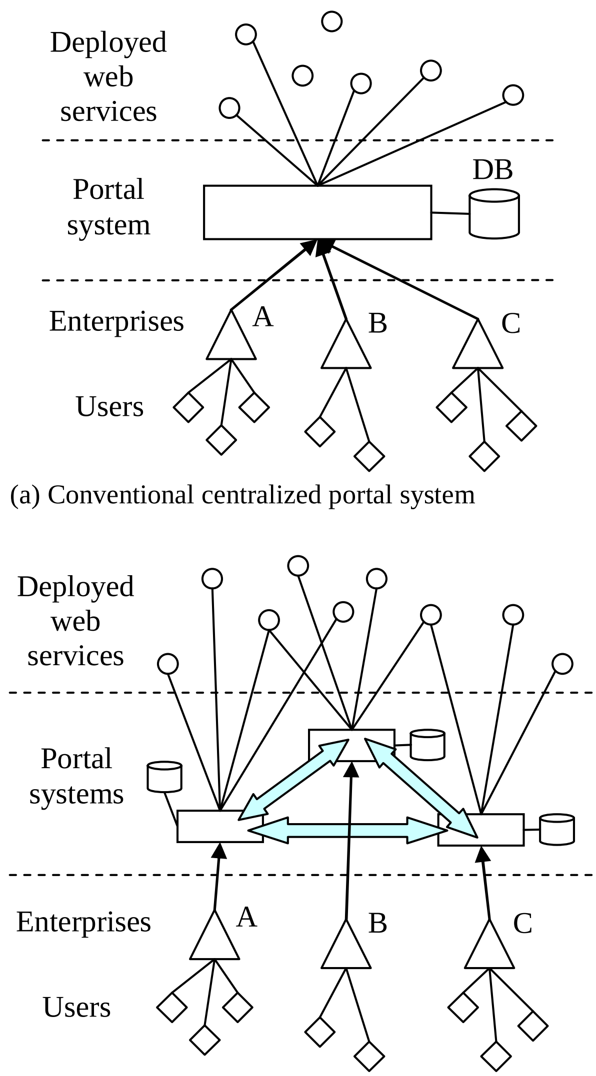
3.1 *Centralized service integration*

Web portal technology has been used to aggregate scattered, distributed information, applications and processes across organizational boundaries. A web portal system provides the clients a single point of access to information and applications regardless of their location or storage mechanism. Through the portal system, multiple applications can be accessed, related and integrated into a workflow or a supply chain. It provides a centralized storage of information and a unified hub to the integrated information, applications and services. Clients can access to multiple systems or applications via the web portal with a single registration and authentication.

The web services model can be implemented using a web portal system. Applications and information sources are wrapped and deployed as individual web portlets, which are web service units that a web portal system can integrate and reuse. Web portlets are sub-programs that encapsulate a single or a number of web applications. They are contained in a portal system and become visible and accessible via the portal system. The sessions and user preferences of each portlet are also stored and managed in the portal system.

3.2 *Decentralized network architecture*

Web portal technology has been deployed by companies for information management and sharing. Web portals are commonly used as content management systems (CMS) and web publishing tools to store digital contents and share them with other project members (Michelinakis 2004). Portal systems therefore act as a centralized repository of information and documents. Some companies build an intranet using web portals. It allows the users to access sensitive internal information and applications, and the administrator to manage a huge amount of information at a centralized location. Both the usages as repository and intranet require a central server and



(a) Conventional centralized portal system

(b) Distributed portal network architecture

Figure 2. Conventional portal usage versus distributed portal network.

database to support the operations of a centralized portal system. This is not practical for supply chain collaboration due to the issues of information ownership rights and proprietary privacy. Companies prefer having their own database system and if necessary, exposing sensitive information such as profit margin only to specific supply chain members at a specific period of time. The use of a central database eliminates the incentive to information sharing across organizations. In this work, a distributed extranet network architecture across organizations is proposed for supply chain integration.

In the distributed network architecture, each organization has its own database and portal system. The portal system can act as an intranet and CMS internally, while at the same time allows information exchange and sharing over the web. As illustrated in Figure 2a, a single portal system is conventionally used to integrate loosely coupled applications and to share information among project participants from different organizations. The database and the portal system are hosted by either one organization or a third party company. With the centralized architecture, individual organizations may hesitate to upload and share their sensitive information depending on their level of trust. On the contrary, the network architecture shown in Figure 2b distributes the storage and ownership of information among enterprises and users. They can grant the rights to view or access their own proprietary data and documents to particular collaborating partners for a specific period of time. The distributed system thus provides better control of the shared information. Enterprises may become more willing to coordinate and share their proprietary information.

4 SC COLLABORATOR

SC Collaborator is a web-based prototype platform that is designed to support interoperation and collaboration among project partners in the construction industry. The platform integrates distributed information sources and engineering services using the web services model. SC Collaborator provides an economical solution to construction supply chain integration by adopting open source technologies. Web portal technology is leveraged to provide the users a customizable user interface. The system implements the distributed portal network architecture discussed in Section 3 with security access control. In the following sections, the system architecture of SC Collaborator will be discussed. Two illustrative scenarios will then be presented to demonstrate the interoperation among general contractor, subcontractors and suppliers using SC Collaborator.

4.1 *System architecture*

Figure 3 shows the system architecture of SC Collaborator. On the server side, SC Collaborator is divided into three tiers – the web server / servlet container tier, the business implementation tier and the database tier. The servlet container tier allows clients to access the system through standard web services protocol by SOAP messaging and WSDL description, through wireless devices by Wireless Markup Language (WML), or through web browsers. The business implementation tier provides connectivity to the database, manages the sessions of the system, manages the information transaction, and performs business functions. The database tier serves as the back-end information source to support the whole platform. It stores information such as user

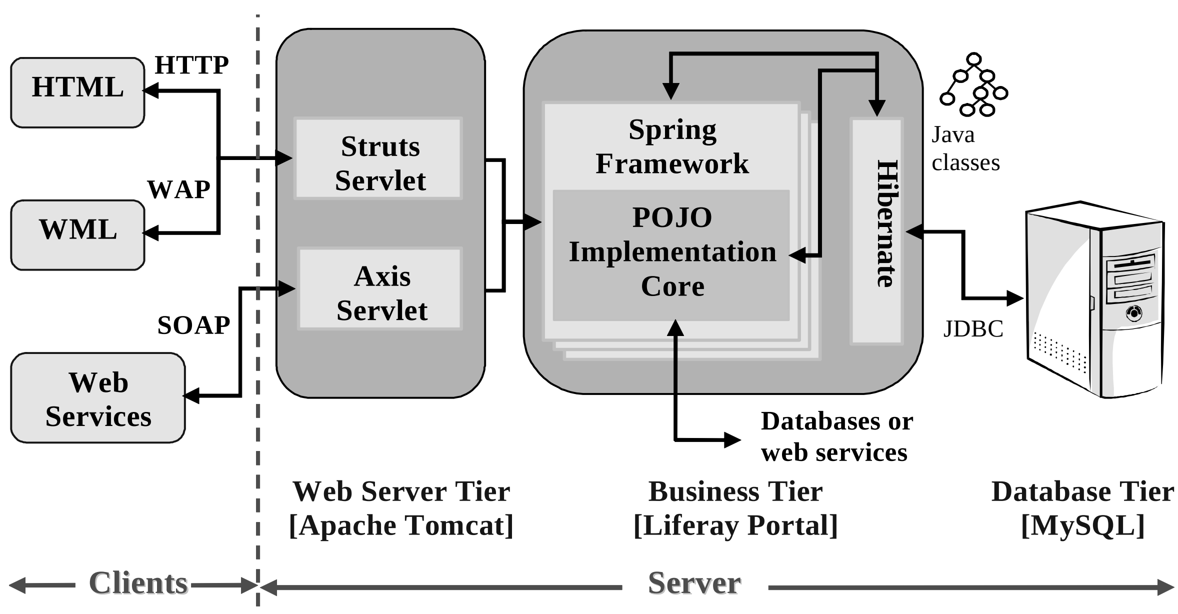


Figure 3. System architecture of SC Collaborator.

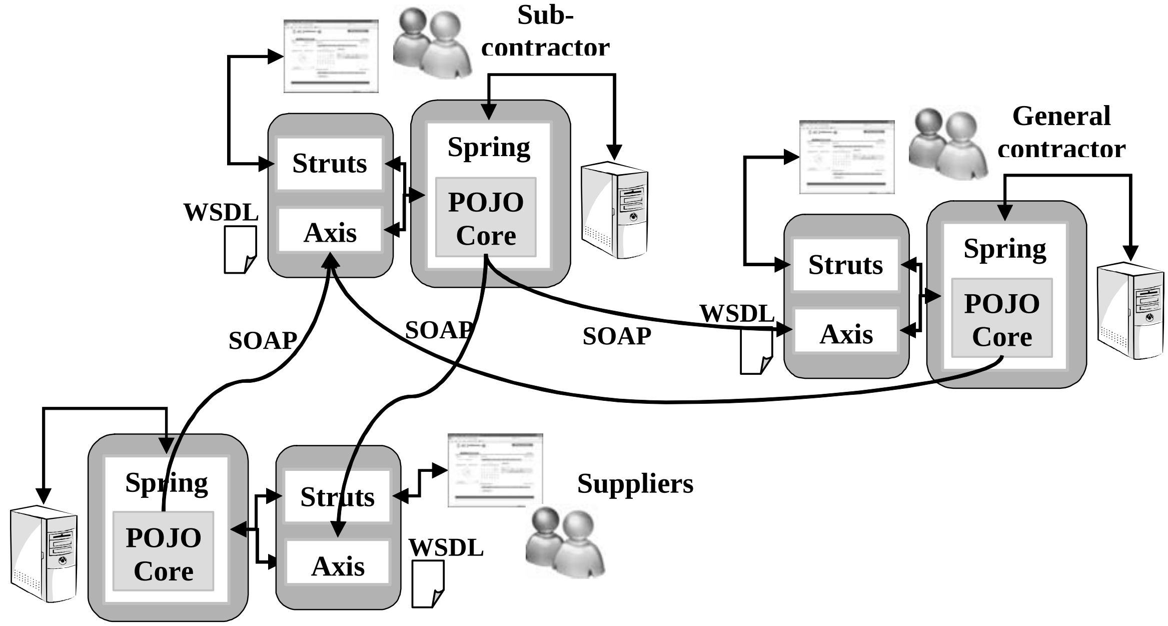


Figure 4. Communications among individual SC Collaborator systems.

privileges and settings, portal and portlet configurations, and page layout settings. The three tiers of SC Collaborator are implemented with open source software – Apache Tomcat, Liferay Portal and MySQL. SC Collaborator provides an economical and desirable platform for the AEC companies, which are usually small to medium in size and are often reluctant to invest on a system that requires frequent changes.

The business tier consists of the Plain Old Java Object (POJO) implementation core and two supporting frameworks, which are all bundled in the Liferay Portal (*Liferay Portal Enterprise* 2007). The POJO implementation core performs the main computational job. The core can connect to other localized applications through APIs or other remote applications via web services. The core is extended with two light-weight frameworks, Hibernate and Spring. The Hibernate framework maps the objects in the relational database into Java object-oriented classes. The Spring framework wraps the POJO core and provides additional features such as messaging, session management and transaction management.

4.2 *Distributed Portal Network*

Each individual SC Collaborator system is a portal platform which can act as a digital content repository or as an intranet managed and hosted by separate organizations. The portal platform provides the clients a single point of access to information sources and applications. A single authentication of the system allows clients to gain access to multiple systems or applications with a unified and customizable user interface. The entire system can be divided into web portlets which can be developed, managed and integrated independently, that makes the system flexible, scalable and easily re-configurable.

SC Collaborator implements the distributed portal network introduced in Section 3. The communication between individual SC Collaborator systems is achieved using standardized web service technologies and languages (Fig. 4). The POJO implementation core extended with the Spring framework supports the invocation of web services by SOAP. The Axis servlet allows portal and portlet functionalities to be exposed and deployed as web services. The deployed functionalities are described using standardized WSDL language for discovery and invocation. Figure 5 shows the WSDL file of a simple portlet functionality which sends purchase orders to suppliers. The connectivity between separate SC Collaborator portal systems can be easily created as long as the address of the deployed web services is given.

Security is an issue that many companies concern for collaborative systems. The portal and portlet functionalities can be deployed as web services in a secure way in SC Collaborator. Successful authentication with correct user ID and password is needed to invoke the web services. The user ID and password share the same profile with the accounts in SC Collaborator. In other words, the system administrator can manage the access rights to the deployed web services by managing the accounts in SC Collaborator using the administrator portlet. The access rights are established or removed when the corresponding SC Collaborator account is created or deleted. This ensures a consistent access control to the portal system as well as the exposed functionalities.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions
    targetNamespace="urn:http.service.material_buyer.portlet.ext.com"
    xmlns:apachesoap="http://xml.apache.org/xml-soap"
    xmlns:impl="urn:http.service.material_buyer.portlet.ext.com"
    xmlns:intf="urn:http.service.material_buyer.portlet.ext.com"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  + <!--   -->
    <wsdl:message name="addPurchaseOrderResponse" />
  - <wsdl:message name="addPurchaseOrderRequest">
      <wsdl:part name="orderId" type="xsd:string" />
    </wsdl:message>
  - <wsdl:portType name="MaterialBuyerServiceSoap">
    - <wsdl:operation name="addPurchaseOrder" parameterOrder="orderId">
        <wsdl:input message="impl:addPurchaseOrderRequest"
          name="addPurchaseOrderRequest" />
        <wsdl:output message="impl:addPurchaseOrderResponse"
          name="addPurchaseOrderResponse" />
      </wsdl:operation>
    </wsdl:portType>
  - <wsdl:binding
      name="Portlet_MaterialBuyer_MaterialBuyerServiceSoapBinding"
      type="impl:MaterialBuyerServiceSoap">
      <wsdlsoap:binding style="rpc"
        transport="http://schemas.xmlsoap.org/soap/http" />
    - <wsdl:operation name="addPurchaseOrder">
        <wsdlsoap:operation soapAction="" />
      - <wsdl:input name="addPurchaseOrderRequest">
          <wsdlsoap:body
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="urn:http.service.material_buyer.portlet.ext.com"
            use="encoded" />
        </wsdl:input>
      - <wsdl:output name="addPurchaseOrderResponse">
          <wsdlsoap:body
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="urn:http.service.material_buyer.portlet.ext.com"
            use="encoded" />
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
  - <wsdl:service name="MaterialBuyerServiceSoapService">
    - <wsdl:port
        binding="impl:Portlet_MaterialBuyer_MaterialBuyerServiceSoapBinding"
        name="Portlet_MaterialBuyer_MaterialBuyerService">
        <wsdlsoap:address location="http://171.67.80.217:8080/tunnel-
          web/axis/Portlet_MaterialBuyer_MaterialBuyerService" />
      </wsdl:port>
    </wsdl:service>
  </wsdl:definitions>
```


Figure 5. Example WSDL file deployed in SC Collaborator.

4.3 *Example Scenarios*

To illustrate the SC Collaborator platform for the applications in the AEC industry, two example scenarios are described in the following sections. In the two examples, distributed SC Collaborator system is installed in each organization. These examples demonstrate the potential of using SC Collaborator to connect distributed web applications and systems, and to act as a unified communication channel among construction project participants.

4.3.1 *Procurement interactions*

E-Procurement has gained huge popularity as many studies have shown its values in supply chain management. Beyond the obvious transaction cost savings and access to suppliers, e-Procurement can offer product standardization, quality assurance, inventory management and the opportunity to manage material flows down the value chain (Issa et al. 2003). As reported by Sanders (2001), the benefits of e-Procurement can be tremendous in terms of savings in cost and time, return on investment, and improvement in customer relationship. This example demonstrates the e-Procurement activities performed using SC Collaborator.

For most e-Procurement systems, it usually takes time to configure and establish the communication channels between buyers and suppliers. On the contrary, the communication channels can be easily and quickly created and removed in SC Collaborator due to its service oriented architecture. To establish a channel, the organizations simply need to create an account in SC Collaborator for their trading partners, and exchange the IP address with each other. This facilitates the addition, replacement and removal of trading partners, which happen frequently in fast changing AEC supply chains.

In this example scenario, an interior designer working with a general contractor company *GenCon* extracts the items (such as furniture items) and associated information from Autodesk Architectural Desktop (ADT) to Microsoft Excel (Fig. 6). The procurement list is submitted to the SC Collaborator system for review. The procurement officer in GenCon can log on SC Collaborator and evaluate the items proposed by the architects. As illustrated in Figure 7, the system stores and keeps track of all the product and status information for each item. By selecting the items and providing the order number ("OH-whx-90" in this case), electronic purchase order can be easily created and sent to the designated suppliers through web services.

The supplier *SuppliOne* logs on its own SC Collaborator system and manages the purchase orders it receives (Fig. 8). The purchase order submitted by GenCon is received in SuppliOne's system instantaneously. After considering its inventory information, SuppliOne decides to offer only two of the requested items and responds to GenCon electronically with a confirmation number. Figure 9 shows the updated information in GenCon's system right after SuppliOne's responses.

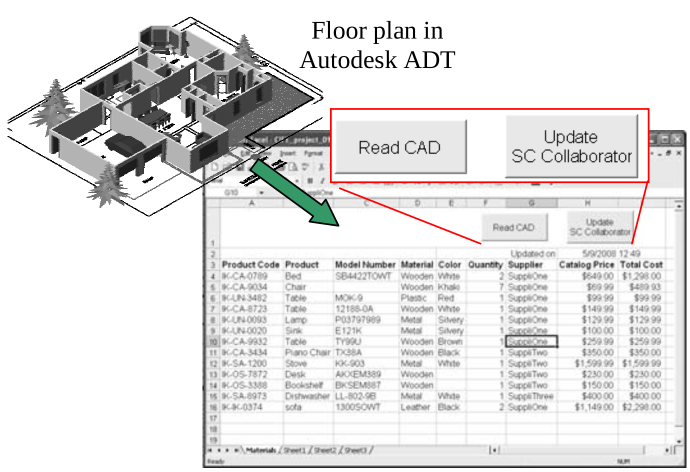


Figure 6. Extracting item information from a floor plan in Autodesk ADT to a spreadsheet in Microsoft Excel.

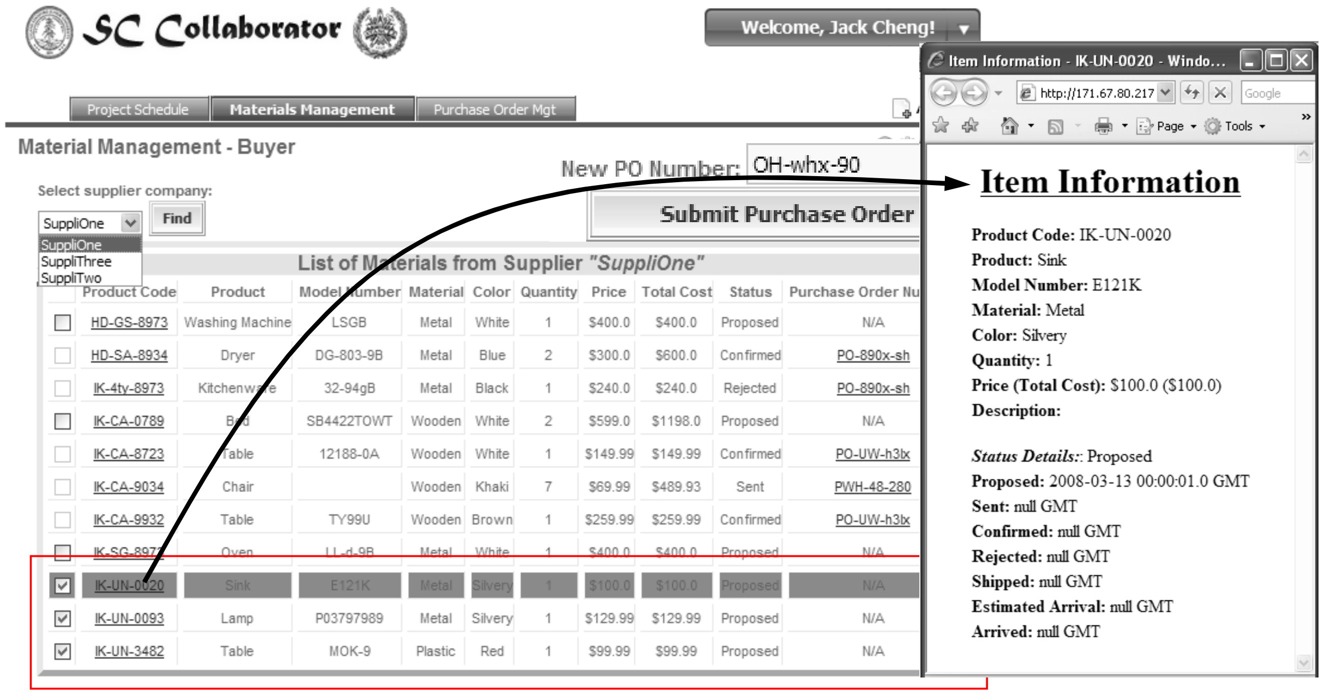


Figure 7. Review of procurement item list and submission of electronic purchase order by GenCon.

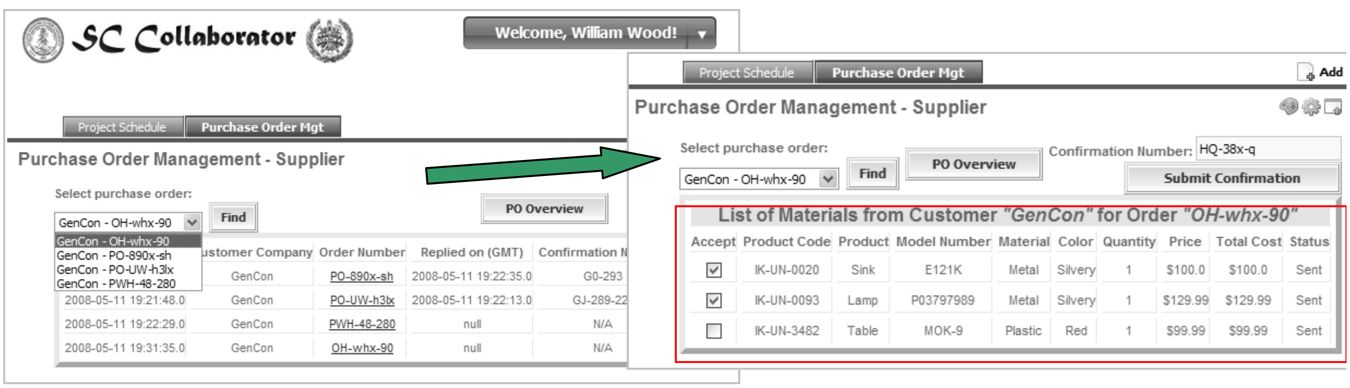


Figure 8. Managing purchase order in SC Collaborator by SuppliOne.

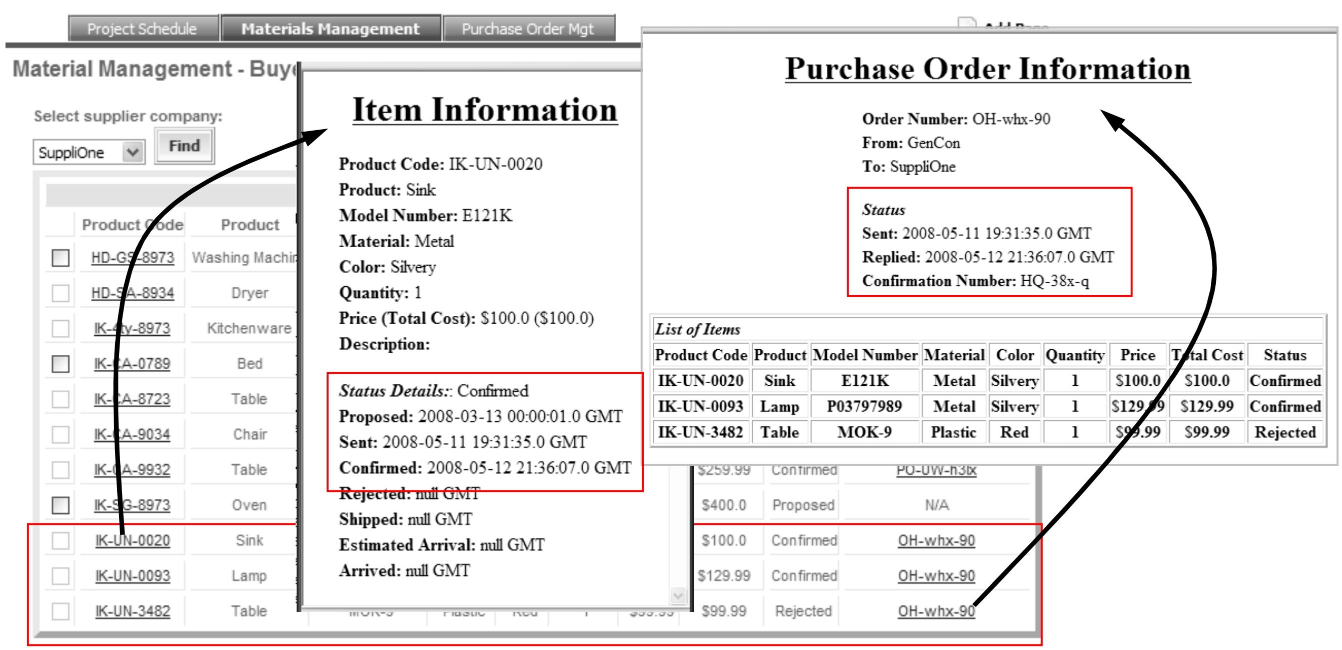


Figure 9. Updated item status and purchase order information in GenCon's SC Collaborator system.

4.3.2 *Material delivery management*

The second example illustrated in Figure 10 demonstrates the online collaboration and information flow among a general contractor, subcontractors and suppliers using SC Collaborator to manage material delivery. In this example, one of the suppliers of the mechanical, electrical and plumbing (MEP) subcontractor has to delay the material delivery of two items. As shown in Figure 10, the delay notice sent by the suppliers triggers a message in SC Collaborator delivered to both the general contractor and the MEP subcontractor.

The general contractor provides the MEP subcontractor with two options – to change the supplier so that the schedule remains unchanged, or to request a task delay with a penalty. The subcontractor uses SC Collaborator to connect to the systems of the suppliers with partnership agreements, and checks the available alternative suppliers for the two items. The subcontractor finds that one of the items is out of stock among the partner suppliers, which means that a task delay is unavoidable. The subcontractor then makes a schedule delay for the affected task in SC Collaborator, as illustrated in Figure 10. All the project participants obtain an updated schedule instantly to plan and revise their tasks.

5 SUMMARY AND CONCLUSION

The lack of interoperability can seriously hinder project coordination and collaboration, which are vital to the success of a supply chain. It has been reported that insufficient interoperability causes huge economic costs to various industries. Therefore, significant values can be added if interoperability is facilitated among supply chain members. Software interoperability which supports the integration of information, software applications and engineering services has been discussed. With the proliferation of Internet, the web services model has emerged as a promising approach to application, process and information integration. With the service oriented architecture, the web services model enables modular system development, reusability of web service components and high re-configurability. Therefore, the web services model can provide a useful framework to facilitate collaboration and interoperation for the fast-changing construction supply chains. Specifically, web portal technologies which implement the web services model are leveraged to provide the users a unified point of access. A distributed portal network architecture is proposed to extend the web portal framework and to promote sharing of

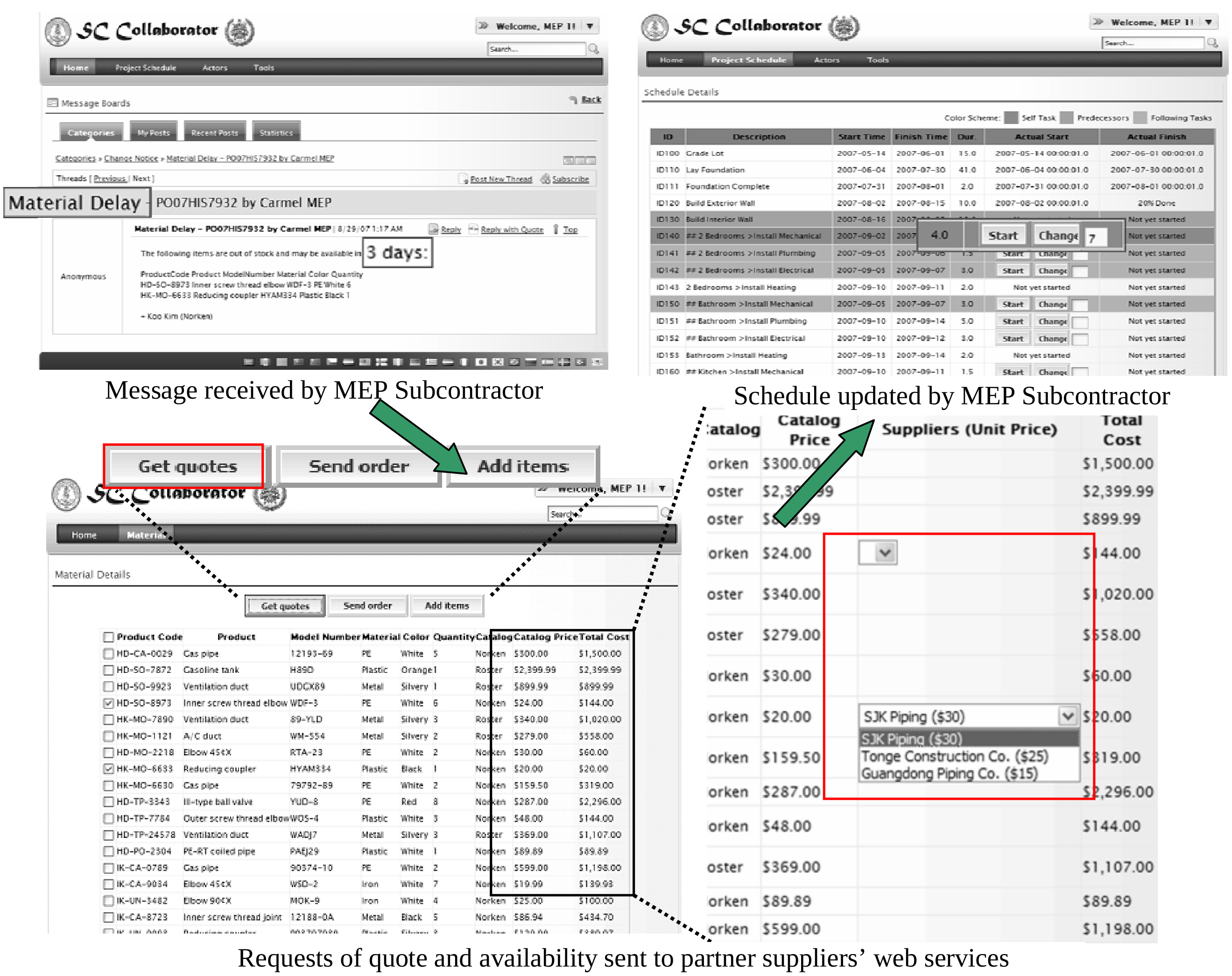


Figure 10. Material delivery management scenario using SC Collaborator.

sensitive and proprietary information.

In this paper, we present SC Collaborator, a prototype web-based portal platform designed to facilitate information sharing and interoperation within and across organizational boundary. The platform implements the distributed network architecture using standardized web services languages and protocols. The platform leverages open source software tools to provide an economical solution for AEC companies, which are mostly small to medium in size and reluctant to invest heavily on IT. Two illustrative examples are provided to demonstrate the interoperation among stakeholders using SC Collaborator for procurement and material delivery in AEC projects.

6 ACKNOWLEDGEMENTS

The authors would like to acknowledge the supports by the US National Science Foundation, Grant No. CMS-0601167, the Center for Integrated Facility Engineering (CIFE) at Stanford University, the Enterprise Systems Group at the National Institute of Standards and Technology (NIST) and Wast-Bygg, AB, Sweden. Any opinions and findings are those of the authors, and do not necessarily reflect the views of NSF, CIFE, NIST or Wast-Bygg, AB. No approval or endorsement of any commercial product by NIST, NSF or Stanford University is intended or implied.

REFERENCES

Andrews, T. et al. 2003. Business Process Execution Language for Web Services (BPEL4WS). *OASIS, Specification Version* 1.

Ankolekar, A. et al. 2001. DAML-S: Semantic Markup for Web Services. *Proceedings of the International Semantic Web Working Symposium (SWWS).*

Anumba, C.J. & Duke, A. 1997. Internet and Intranet Usage in a Communication Infrastructure for Virtual Construction Project Teams. *Proceedings of the 6th IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*: 56-61.

Begley, E.F. et al. 2005. *Semantic Mapping Between IAI ifcXML and FIATECH AEX Models for Centrifugal Pumps.* Technical Report, NIST.

Bingi, P. et al. 2001. Critical Issues Affecting an ERP Implementation. In J.M. Myerson *Enterprise Systems Integration*: 425-438. Auerbach Publications, CPC Press.

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

Brucker, B.A. & Nachtigall, S.D. 2005. Initiating The Building Information Model From Owner and Operator Criteria and Requirements. *International Conference on Computing in Civil Engineering*.

Brunnermeier, S.B. & Martin, S.A. 2002. Interoperability Costs in the US Automotive Supply Chain. *Supply Chain Management: An International Journal* 7(2): 71-82.

Cheng, C.P. et al. 2008. Domain-Specific Ontology Mapping by Corpus-Based Semantic Similarity. *Proceedings of 2008 NSF CMMI Engineering Research and Innovation Conference,* Knoxville, Tennessee, USA.

Cheng, E.W.L. et al. 2001. An E-business Model to Support Supply Chain Activities in Construction. *Logistics Information Management* 14(1/2): 68-77.

Cheng, J. 2004. *A Simulation Access Language and Framework with Applications to Project Management*, Ph.D. Thesis, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA.

Cheng, J. et al. 2006. Engineering Simulations with Web-based Services. *Proceedings of the 11th International Conference on Computing in Civil and Building Engineering, (ICCCBE XI)*, Montreal, Canada.

Cheng, J. et al. 2002. Ontology Mapping Between PSL and XML-Based Standards For Project Scheduling. *Proceedings of the Third International Conference on Concurrent Engineering in Construction*, Berkeley, CA, USA: 143-156.

Danso-Amoako, M.O. et al. 2006. Developing a Framework to Support Data Exchange from Heterogeneous Sources via IFC and Web Services. *Proceedings of the 11th International Conference on Computing in Civil and Building Engineering, (ICCCBE XI),* Montreal, Canada: 2477-2486.

de Bruijn, J. et al. 2005. Web service modeling ontology (WSMO). W3C Member Submission. *World Wide Web Consortium (W3C).*

Gallaher, M.P. et al. 2004. Cost Analysis of Inadequate Inoperability in the Capital Facilities Industry. *National Institute of Standards and Technology (NIST) Technical Report No. NIST GCR-04-867, August*.

Greenwood, D. et al. 2004. Engineering Web Service-Agent Integration. *2004 IEEE International Conference on Systems, Man and Cybernetics* 2.

Gunasekaran, A. & Ngai, E.W.T. 2004. Information Systems in Supply Chain Integration and Management. *European Journal of Operational Research* 159(2): 269-295.

International Alliance for Interoperability 1997. *Guidelines for the Development of Industry Foundation Classes (IFC).* IAI.

Ismail, A. et al. 2002. When Computers Learn to Talk: A Web Services Primer. *The McKinsey Quarterly*.

Issa, R.R.A. et al. 2003. A Survey of E-business Implementation in the US Construction Industry. *Journal of Information Technology in Construction* 8: 15-28.

Lee, H.L. et al. 2004. Information Distortion in a Supply Chain: The Bullwhip Effect. *Management Science* 50(12 Supplement): 1875-1886.

Lee, H.L. et al. 2000. The Value of Information Sharing in a Two Level Supply Chain. *Management Science* 46(5): 628-643.

Lee, H.L. & Whang, S. 2005. Supply Chain Integration over the Internet. In *Supply Chain Management: Models, Applications, and Research Directions*: 3-17. Springer US.

Lee, J. & Bernold, L.E. 2008. Ubiquitous Agent-Based Communication in Construction. *Journal of Computing in Civil Engineering* 22(1): 31-39.

Leymann, F. 2001. Web Services Flow Language. *IBM Software Group*.

Liferay Portal Enterprise, 2007. http://www.liferary.com/.

Lipman, R. 2006. Mapping between the CIMsteel Integration Standards (CIS/2) and Industry Foundation Classes (IFC) Product Models for Structural Steel. *Proceedings of ICCCBE*, Montreal, Canada: 3087-3096.

Manthou, V. et al. 2004. Virtual E-Chain (VeC) Model for Supply Chain Collaboration. *International Journal of Production Economics* 87(3): 241-250.

Michelinakis, D. 2004. *Open Source Content Management Systems: An Argumentative Approach*, Master Thesis, Warwick Manufacturing Group, University of Warwick.

Preist, C. 2004. A Conceptual Architecture for Semantic Web Services. *Proceedings of the 3rd International Semantic Web Conference (ISWC 2004)*: 395-409.

Ray, S.R. 2002. Interoperability Standards in the Semantic Web. *Journal of Computing and Information Science in Engineering* 2(1): 65-69.

Sanders, M.R. 2001. What Does eMarketplace Buying Cost. Tech Strategy, Forrester Research.

Watson, A. & Crowley, A. 1995. CIMSteel Integration Standard. In Product and Process Modelling in the Building Industry: 491-493. Rotterdam: A.A. Balkema.

Yu, Z. et al. 2001. Benefits of Information Sharing with Supply Chain Partnerships. Industrial Management and Data Systems 101(3): 114-119.