CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

Construction Supply Chain Visualization Through Web Services Integration

By

Jung Ung Min & Hans Bjornsson

CIFE Technical Report #149
MAY 2004

STANFORD UNIVERSITY

COPYRIGHT © 2004 BY
Center for Integrated Facility Engineering

If you would like to contact the authors, please write to:

c/o CIFE, Civil and Environmental Engineering Dept.,
Stanford University
Terman Engineering Center
Mail Code: 4020
Stanford, CA 94305-4020

Construction Supply Chain Visualization Through Web Services Integration

Jung Ung Min[1] and Hans C. Bjornsson[2]

Abstract: This paper presents *SCVisualizer*, a real-time information visualization tool for construction supply chain management. *SCVisualizer* leverages Web Services and computer agents technology for building a virtual construction supply chain. Having real-time information available at any time can increase information transparency, which in turn, makes it easier for project managers to identify potential risks. However, this is not an easy task since this information has to be retrieved from heterogeneous systems. In addition, the process of information retrieval is intractable for practitioners. In order to increase information transparency in construction supply chain management, this paper introduces a prototype system that constructs a virtual supply chain seamlessly. A hybrid approach for combining the existing two technologies is also proposed for ensuring the connectivity and the interoperability of software components over the Web.

Keywords: construction, web services, computer agent, supply chain management

---

[1] Dept. of Civil and Environmental Engineering, Stanford University, E-mail: jumin@stanfordalumni.org

[2] Dept. of Civil and Environmental Engineering, Stanford University, E-mail: hansbj@stanford.edu

1. Introduction

This paper introduces *SCVisualizer (Supply Chain Visualizer)*, an information visualization tool using Web Services and computer agent technology for the rapid and seamless generation of a virtual supply chain in construction. The supply chain can significantly affect a construction project based on its ability to deliver construction products in an efficient and timely manner. Recently, supply chain management has received a lot of attention from the construction industry.

Traditionally, project management in construction follows an activity-centered approach that concentrates on monitoring project participants' activities against a construction schedule (Howell 1999). Since the underlying motivation of this approach is to improve efficiency in value-adding activities where physical conversions occur (Koskela 1992), passive communications from downstream (customer side) to upstream (supplier side) have been prevailing in the industry. However, due to the long information lead-time and the lack of coordination, the initiated communication often fails to have the required information in a timely manner. To make matters even worse, the communication itself may not be initiated because of the large number of project participants in a construction project: it is an overwhelming task for project managers to cover numerous project members and their suppliers. These types of disturbances in information flows generate turbulences in material flows, which are one of the highly ranked causes of delay in a construction project. Thus, the coordination of information flows is a key component in achieving tight integration in order to optimize the chain-wide performance.

As a way of coordinating information flows, the idea of real-time information sharing has been studied extensively in the manufacturing industry. It specifically emphasizes instantaneous multilateral information sharing within a supply chain in order to reduce uncertainties associated with operations and demand forecast (Cooper et al. 1997; Lewis and Talalayevsky 1997). Recently, this idea has been facilitated due to the rapid advances in information technology (Lee and Whang 2000). Among these cutting-edge technologies, Web Services is an emerging technology that many people envision its applicability to supply chain management (Castro-Leon 2002; Preist et al. 2001; Tosic et al. 2001). It can be used for building and integrating distributed applications within and across corporate boundaries relying upon the existing and emerging standards. Considering the problems of activity centered approach in project management and the features of Web Services technology, this technology could provide agility as well as flexibility to construction supply chain management, where it deals with the complex and sequential flows of materials and information. This is because the core concept of Web Services technology is ensuring the connectivity and the interoperability of software components over the Web. However, since it has not been developed only for construction, it is not obvious how this technology can specifically support supply chain management in construction.

In contrast to the manufacturing industry, the construction industry has a wide spectrum of speed, uncertainty, and complexity (Ballard and Howell 1998). Introducing existing strategies and techniques might not be suitable for this industry (Ballard and Howell 1998). Therefore, it is necessary to assess Web Services technology and its applicability from a construction-specific point of view. In particular, this paper focuses the expediting process, where a project manager (or an expeditor) will monitor the vendor's progress to make sure that the ordered products or components are delivered by the agreed date.

2. Supply Chain Management in Construction

Information plays a very important role in construction management in various ways. Many researchers emphasized the importance of communication and information exchange between project participants in the material flow control process (Agapiou et al. 1998; Vrijhoef and Koskela 1999). In this context, numerous researchers have focused on sharing information in construction supply chain management.

Nicolini et al. (2001) pointed out the inefficiency of centralized coordination system in managing interdependencies within a construction supply chain. In order to facilitate information sharing, they suggested a cluster, which is a temporary organization consisting of designers and suppliers, to support intensive collaboration between different disciplines. Clustering design helps minimize interfaces, which in turn, facilitates communication transparency (Nicolini et al. 2001). In terms of this type of collaborative design, Bogus et al. (2000) claimed that design team should be expanded such that it includes contractors, subcontractors, and material suppliers. They described that traditional constructibility concepts have contributed to improve the flow of construction process but they should not be limited only to designers and contractors. However, they expected that communication will be the most difficult part but recent advances in information technology could make it easier for them to communicate (Bogus et al. 2000). Arbulu and Tommelein (2002) also stressed the importance of coordination and communication between the participants in the supply chain using the design of pipe supports as an example. They claimed that in practice, real-time feedback and pull of materials to the site seemed to be missing in this design process (Arbulu and Tommelein 2002).

In addition to the information sharing in the design phase, Chua et al. (1999) and Choo and Tommelein (2000) extensively discussed the planning and the scheduling perspective of information sharing. In the distributed Integrated Production Scheduler (IPS) model, all the members of a project are responsible for executing schedules and providing relevant information so that all the process is clearly visible to the others (Chua et al. 1999). Choo and Tommelein (2000) emphasized the importance of well-structured communication and coordination in a dynamic and complex project. They developed a database program called WorkMovePlan in order to automatically create lookahead plans and weekly work plans (Choo and Tommelein 2000). Both of the approaches allow project participants to share the latest schedule information and to propagate conflicts if there is any.

Using Last Responsible Moment (LRM) framework, Lane and Woodman (2000) tried to improve the flexibility of information transfer in the delivery process. LRM is a information transfer mechanism, which is applying the concept of JIT to the information flow (Lane and Woodman 2000). Contrary to facilitating information sharing within the given process, Vaidyanathan (2002) redesigned the information flows in the house reconstruction market from a large retailer's perspective. In his framework, the retailer takes control of information distribution in order to enhance the accountability to the end customer (Vaidyanathan 2002). Similarly, Taylor and Bjornsson (2002) proposed the "e-chain" framework to have benefits from information and material flows. They also pointed out that business must be willing to share key data, and disparate legacy systems have to be integrated, to exchange information seamlessly (Taylor and Bjornsson 2002). For integrating heterogeneous legacy systems in construction, O'Brien et. al (2002) proposed SEEK (Scalable Extraction of Enterprise Knowledge) toolkit. SEEK provides structured approaches to integrate semantically heterogeneous information using wrappers and access layer architecture (O'Brien et al. 2002).

Although many studies have shown the various causes of problems in construction supply chain and different approaches to remedy them, it is difficult to find any efforts of describing detailed approaches for securing real-time information in construction supply chain management. Further, none of them has mentioned the issue of interoperability, which is a critical problem in implementing their solutions in practice. A construction supply chain is a dynamic supply chain where the customers' needs and requirements are continuously changing, as in military logistics (Simon 2001). In this case, rapid deployment is more important than optimization, and information technology is needed to make information visible in real-time as they occur (Kumar 2001).

3. Web Services and Computer Agents

In this paper, the term "Web Service" specifically means a service available over the Web that uses open standards (such as SOAP and WSDL). The importance of SOAP and WSDL is that they offer a simple standard for reading input and output messages a service receives and responses over a standard transport (Petrie 2003, p 4). Fortunately, SOAP and WSDL have been adopted rapidly by industry in general and have become major standards for such Web Services (Petrie 2003).

Since a Web Service has a machine-readable description and SOAP allows loose-coupling, in principle it is possible to use arbitrarily complex software programs to read WSDL and use SOAP to invoke services by exchanging messages (Petrie 2003). However, in reality, there are problems with WSDL: it doesn't specify the sequence of operations (or required services) that can control the integration of services and processes. In a business process such as construction supply chain management, the capability of orchestrating a complex process is an essential component for the automation of the process. This stimulated the idea of integrating Web Services and computer agents. Computer agents can have intelligence that allows them to form architecture to solve complex problems.

In combining Web Services and computer agents, a couple of different approaches have been studied. One of these is the Semantic Web. The Semantic Web extends the current Web to make the Web more accessible to computer agents. In this framework, computer agents leverage well-established ontology languages such as DARPA Agent Markup Language (DAML [3]) (Berners-Lee et al. 2001; Huhns 2002; McIlraith et al. 2001). DAML represents its ontology using the Resource Description Framework (RDF [4]) schema and XML syntax to support the exchange of knowledge. DAML allows computer agents to understand the content of a Web page so that a user can retrieve information systematically from the millions of information sources on the Web. From a service's point of view (compared to DAML's static information retrieval), DAML Services (DAML-S[5]) was developed as a DAML-based Web Services ontology. DAML-S has the capability of expressing a Web Service and its description in a explicit way so that computer agents can understand the meaning of services (Martin et al. 2003). This way of combining Web Services and computer agents extends the WSDL specification such that DAML-S directly combines with WSDL as an ontology definition and the computer agents which have the capability of understanding DAML consume Web Services. However, this approach uses a complex procedural language which is very difficult to implement and deploy, as with the WSFL and XLANG (Ambroszkiewicz 2002). This research seeks to leverage existing Web Services standards in terms of a specific supply-chain management process in construction. For this purpose, the DAML-S approach seems too redundant and heavy-weight because the expediting process is relatively well-defined with limited ontology variation.

4. *SCVisualizer*: Supply Chain Visualization Tool for Construction

SCVisualizer is an information visualization tool that leverages Web Services and computer agent technology for constructing a virtual supply chain seamlessly. In order to bridge the gap between the existing Web Services standards and computer agents technology, this paper introduces a modified WSDL (Web Services Description Language) specification, called AWSDL (Agent Web Services Description Language), which provides interfaces between WSDL and computer agents. This is described in detail in Section 4.2.

4.1 AWSDL Agents in *SCVisualizer*

AWSDL agent is a type of software agent, which can interact over the AWSDL interface described in 4.2. In short, an AWSDL agent has the capability of exchanging ACL (Agent Communication Language) messages within WSDL specification. The current implementation of *SCVisualizer* includes three different types of AWSDL agents: General Contractor

[3] http://www.daml.org/

[4] http://www.w3.org/RDF/

[5] http://www.daml.org/services/

(GCA), Subcontractor (SUBA), and Supplier AWSDL agents (SUPA). Their behaviors differ slightly from one another, reflecting the current business practice in construction. The developed *SCVisualizer* currently keeps one general contractor, ten subcontractors, and ten suppliers.

In *SCVisualizer*, an AWSDL agent's service transaction (request) appears as a plain Web Service request in the form of normal WSDL. Once this arrives, AWSDL invokes the recipient AWSDL agent. Each AWSDL agent has a list of legal methods (such as sending and receiving a message specification) and the operation of basic behavior (retrieving requested information from another source) that is specified beforehand. In addition, each AWSDL agent in *SCVisualizer* maintains three key components: a set of structured data sources, a reasoning engine, and a knowledge base. The following description explains these components with the GCA as an example.

For its structured data set, the GCA accesses a construction schedule and directory data that are stored in the AWSDL Server. Both sets of data have been exported to Microsoft spreadsheet files and the GCA reads and imports them as an object. With the schedule data, the GCA utilizes its reasoning engine to construct its knowledge base. For example, the GCA extracts critical path, predecessor and successor information, and scheduled progress data from raw construction schedule data. Directory data contain the information of other agents' end-point locations (service location as a URL format) as well as their relationship (e.g., Subcontractor A's direct supplier is Supplier B). The raw data are prepared initially by a project manager. During the actual expediting process, the GCA gathers all the relevant information from the other participating agents in real-time. It includes actual progress information, order status, delivery status, and production lead-time. Once the GCA receives information from SUBAs or SUPAs, it will add this to its knowledge base as an expediting report object. The GCA then uses its knowledge base and reasoning engine to identify conflicts or other problems and to plan its actions. In *SCVisualizer*, the GCA sends a warning signal to a human user when it identifies a problem. Also, it selectively propagates the information among the members of the supply chain. Although the current version of *SCVisualizer* gathers and analyzes a limited amount of information, this can be expanded easily. The AWSDL agent model facilitates seamless interconnectivity in a distributed environment. All that we need is to determine the types of information and then to instruct agents how to read and use it.

4.2 Bridging the Gap between WSDL and Computer Agents: AWSDL

4.2.1 AWSDL Specification

Basically, the AWSDL (Agent Web Service Description Language) framework utilizes the WSDL standard. AWSDL provides a vehicle for computer agents to exchange information during the integration of Web Services. It includes regular WSDL services and connects to the special WSDL interface, which describes behaviors (methods) of computer agents.

Since AWSDL supports message exchange between computer agents, AWSDL is classified as a communication layer as shown in Figure 1.

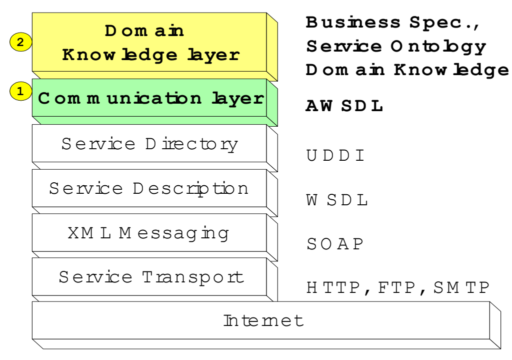


Figure 1: Web Services Stack with AWSDL

This layer deals with communication functions during the actual Web Services integration (such as transactions and coordination between agents), which are not supported by other lower layers but required for computer agents. In terms of positioning the AWSDL specification among other Web Services standards, it is located on top of WSDL and UDDI but lower than the domain knowledge layer. The domain knowledge layer defines the knowledge of business processes, service ontology, and relevant functions that computer agents need to maintain. In *SCVisualizer*, which will be described in Section 5, this layer contains construction schedule information, expediting process information and its ontology, project member information, and etc.

AWSDL describes the interface specification of computer agents. Since AWSDL is extended on top of current WSDL specification, it is fully compatible with other regular WSDL Web Services. The original WSDL specification has six major elements as shown in Figure 2. Among these elements, *types* and *messages* elements were modified so that computer agents can consume published Web Services. In terms of modified *types* element, an *ACLMessage* type was defined which is designed to provide the general-purpose communication specification between computer agents. It is based on FIPA ACL message structure specification [6] and serves as a basic conversational protocol between agents. In *messages* element, two different messages were defined: *serviceRequest* and *invokeMessage*. The *serviceRequest* message is designed to be consumed by human users. For example, if a user needs to initiate the expediting process, he can request the service by invoking *serviceRequest* message with "expediting service" as a string argument. On the other hand, the *invokeMessage* can be consumed only by computer agents. Whenever an agent sends request (or response) to another agent, it needs to invoke *invokeMessage* with *ACLMessage* as an argument. Therefore, the agent has to compose request/response message with the *ACLMessage* type. Figure 3 shows actual AWSDL document that the general contractor agent uses in *SCVisualizer*.

4.2.2 AWSDL Server and Agent Model

The AWSDL Server controls the communications between computer agents using AWSDL as well as the original WSDL specification. Specifically, the AWSDL Server manages the conversations between computer agents as a part of Web Services integration.

[6] http://www.fipa.org/specs/fipa00061/SC00061G.html

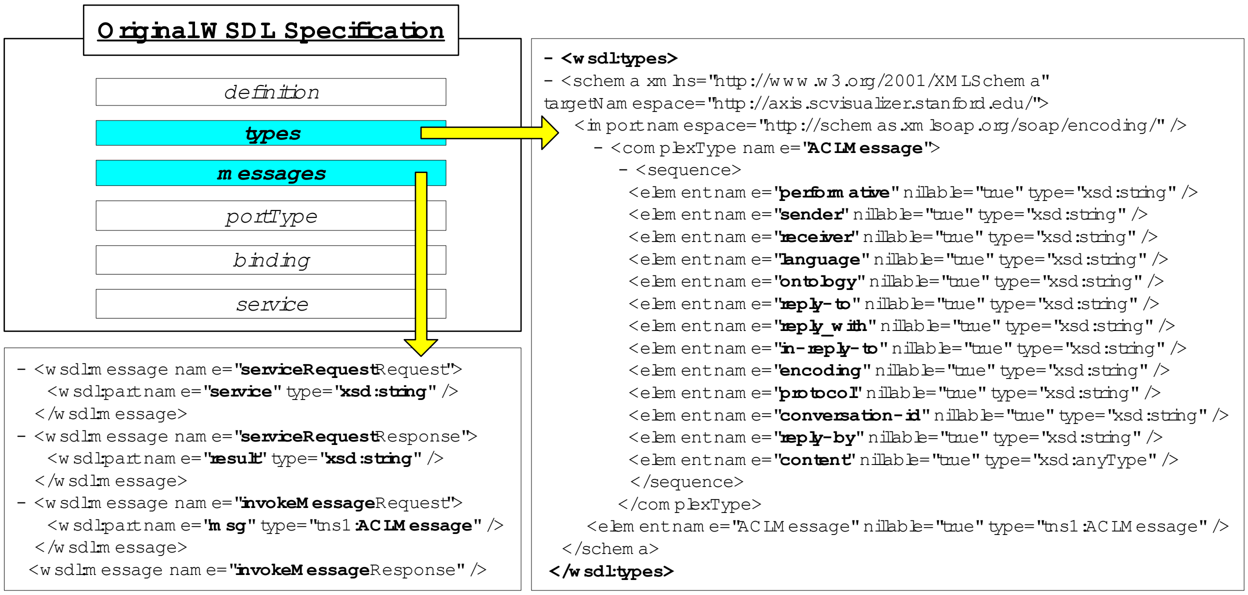


Figure 2: AWSDL Specification

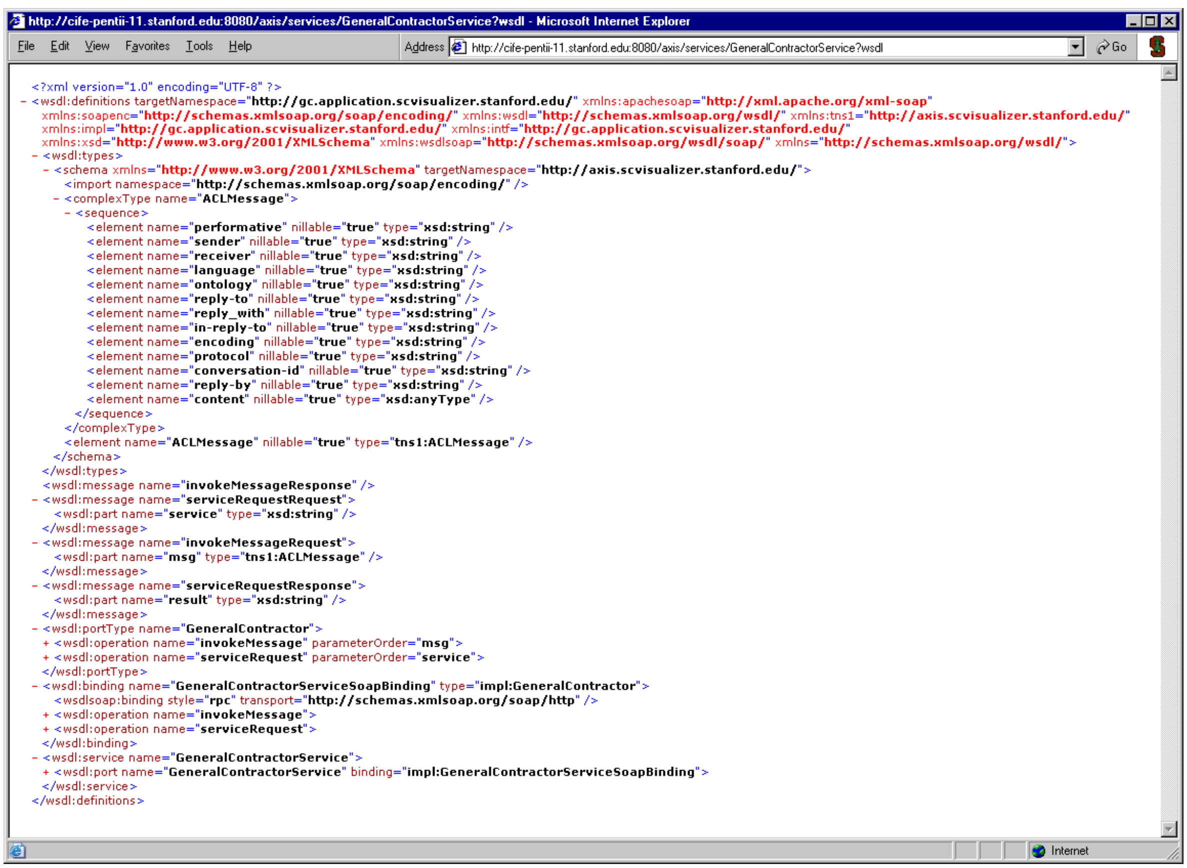


Figure 3: Example of AWSDL Specification (General Contractor Service)

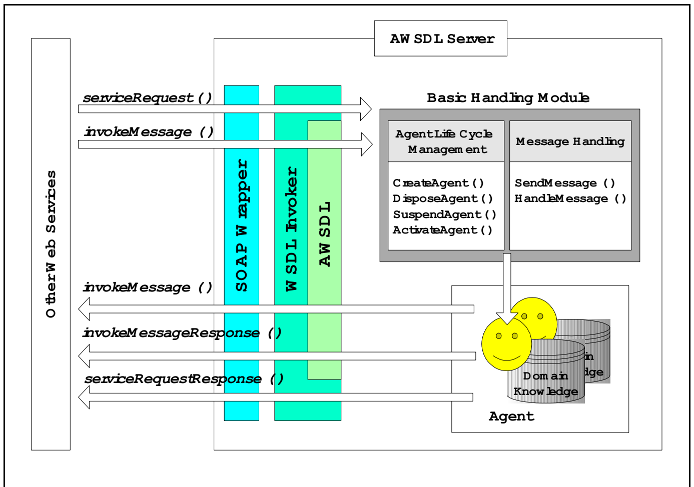


Figure 4: System Architecture of AWSDL Server

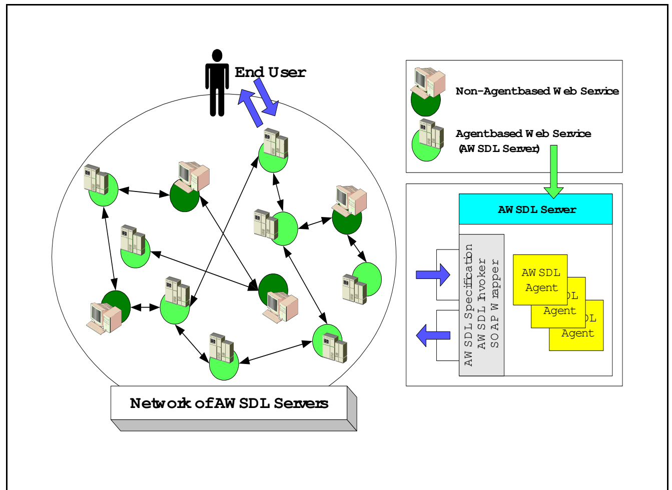


Figure 5: Distributed AWSDL Agent Model

During the actual service process, the AWSDL Server will process requested services by invoking recipient computer agents. The AWSDL Server can deploy multiple agents. For example, in *SCVisualizer*, one general contractor agent and 20 subcontractor/supplier agents were populated. Figure 4 shows the system architecture of the developed AWSDL Server.

The AWSDL Server consists of the following four different modules: SOAP wrapper, WSDL invoker, Basic Handling Module (BHM), and computer agents. In order to invoke a service from a remote machine, all the service requests should be delivered via SOAP so that machines can initiate the requested service procedure.

In order to differentiate agents spawned from AWSDL Server explicitly, these agents were defined as "AWSDL agents," which indicates that these computer agents run on top of the AWSDL environment. With the previously described AWSDL framework, we can construct an independently distributed network of AWSDL Servers as shown in Figure 5.

Given the assumption that a standardized set of ontology exists, an AWSDL agent can cooperate not only with other AWSDL agents but also with other non-agent Web Service interface (a regular WSDL interface). This is because the interface of an AWSDL agent is based on the WSDL specification. Therefore, Web Services can be integrated regardless of the presence of computer agents as well as of platform-specific requirements. This is especially important for construction supply chain management because construction is a highly fragmented industry and a supply chain consists of project-based temporary participants. That is, a virtual supply chain of a construction project can be generated rapidly without concerning the interoperability issue. In addition, having the capability of consuming non-agent Web Services provides an opportunity to integrate all the available Web Services resources – seamless connection to financial, tracking, and procurement Web Services.

5. *SCVisualizer* Demonstration

This section demonstrates the independently distributed framework and *SCVisualizer* in three perspectives: the expediting process, the complex interaction, and the integration with non-agent based Web Services.

5.1 Demonstration of Expediting Process

Figure 6 shows the graphical user interface of *SCVisualizer* in the expediting process module. In the figure, the general contractor agent (GCA) presents three different types of information, which specifically correspond to the three components described in previous section. On the left side, it shows the information of the supply chain. This information is restructured from a directory data source to represent a subcontractor-supplier relation explicitly. On the right side, the first table shows an expediting report. Using the directory and the information of subcontractor-supplier relationship, the GCA gathers relevant information from 20 different project participants and displays it for a project manager.

Figure 7 illustrates actual SOAP messages that the general contractor agent exchanges with other agents during this information gathering process. Request and response ACL messages are transferred as a SOAP payload. With reasoning engine and knowledge base, the GCA identifies problems and chooses appropriate actions. As shown in the Figure 6, the GCA gives an early warning to the project manager. Although it cannot be seen from the figure, the GCA sends an email alert to the project manager. At the same time, the GCA selects directly affected subcontractors and suppliers using its reasoning engine and propagates the information to them in real-time.

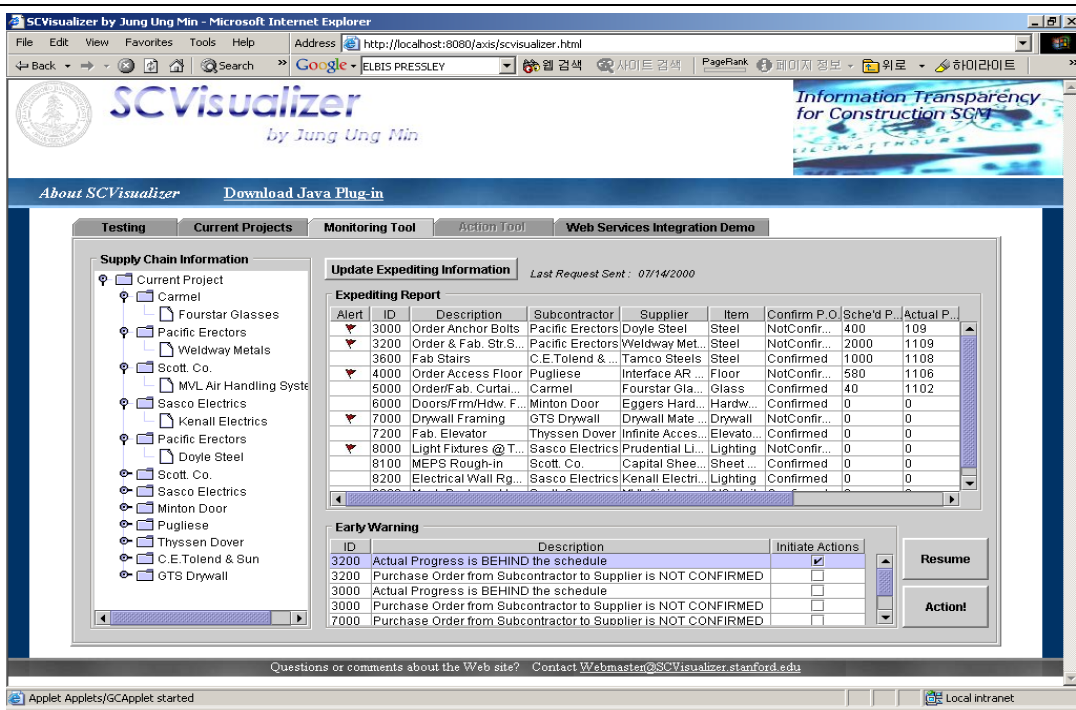
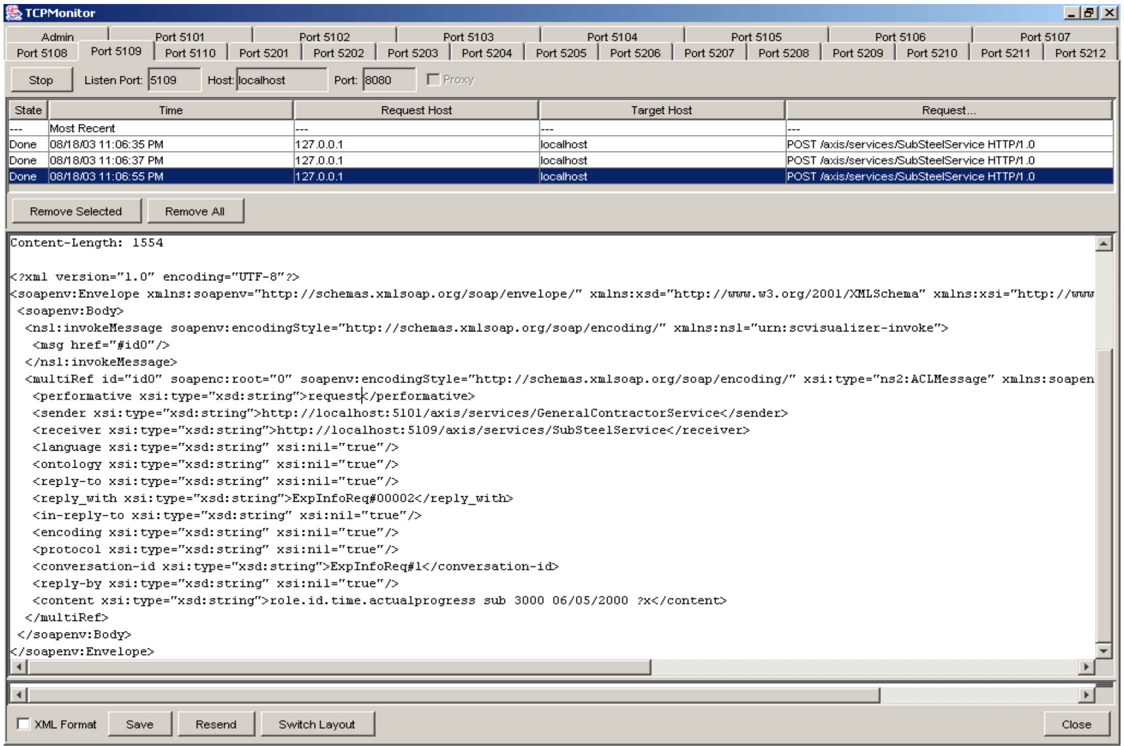
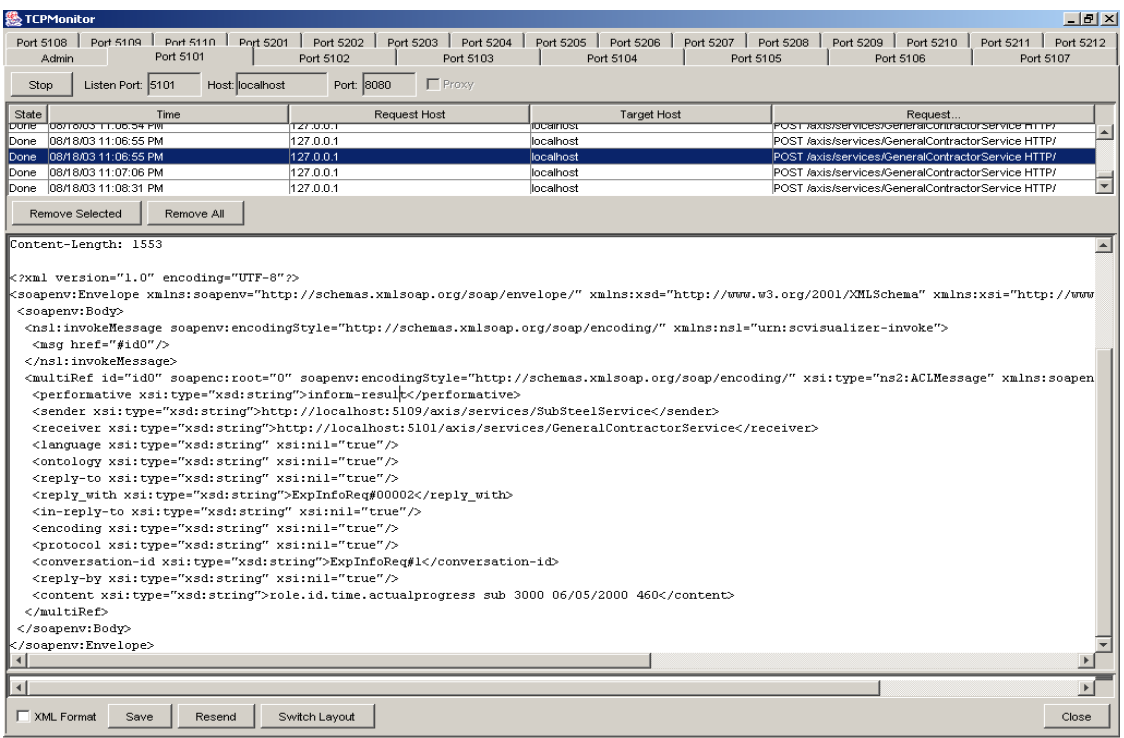


Figure 6: *SCVisualizer* – Expediting Process Demonstration



SOAP Request Message



SOAP Response Message

Figure 7: *SCVisualizer* – Actual SOAP Messages

5.2 Demonstration of Complex Interaction

This demonstration shows how *SCVisualizer* can support more complex interactions with other agents. Once a project manager identifies a potential or existing problem, he can delegate to the general contractor agent (GCA) in order to resolve the situation. The negotiation is based on the concept of option theory: members of a construction supply chain trade the real-time information that is monitored by *SCVisualizer*.

The basic philosophy of this negotiation is that if real-time information pertaining to subcontractors' (or suppliers') progress makes an option contract attractive to some project participants, the general contractor sells this information to them in the form of an option. The general contractor can mitigate its imminent risks by having the option premium. At the same time, the option holders might have a chance to get additional benefits. For this purpose, FIPA Contract Net Interaction Protocol Specification [7] was used as an interaction protocol. Once the GCA is deployed for the negotiation, it will send CFP (Call-for-Proposal) to all the associated agents in the construction schedule. With the received proposal, the GCA suggests a solution based on the project manager's decision criteria. Figure 8 shows an example of a solution where activity 3200 has a problem which might delay the project. The project manager identified the problem from *SCVisualizer* and initiates negotiation. Before initiating the negotiation, the GCA classifies activities using its knowledge base and reasoning engine into

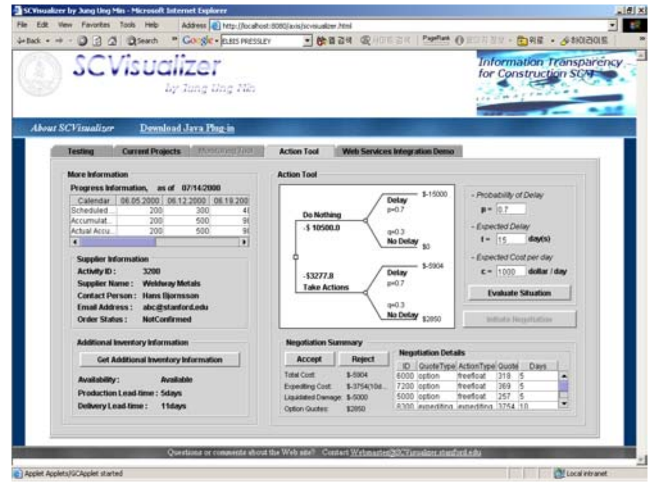


Figure 8: *SCVisualizer* – Negotiation Demonstration

two different categories: critical and non-critical. Based on this classification, the GCA sends different types of CFP to the other agents. For non-critical activities, subcontractor agents submit an option quote that allows them to use free float time (since all the construction work is legally bounded contract, a subcontractor needs to finish its work by the agreed date, which might not be necessary in terms of the overall construction schedule). For critical activities, subcontractor agents can submit two different options: expediting quote and delay quote. The expediting quote is the price that needs to accelerate its activity which will be paid by the general contractor. The delay quote is option quote that the subcontractor is willing to pay in return for some additional buffer time.

In the figure, the GCA suggests a solution as follows:

- Accept ID 6000's option proposal & receive 318 units for the 5 days of free float time.

- Accept ID 7200's option proposal & receive 369 units for the 5 days of free float time.

[7] http://www.fipa.org/specs/fipa00029/SC00029H.html

- Accept ID 5000's option proposal & receive 257 units for the 5 days of free float time.

- Accept ID 8300's expediting proposal & pay 3754 units for expediting 10 days.

5.3 Demonstration of Integration with Non-Agent Web Services

This section demonstrates how *SCVisualizer* can integrate multiple non-agent based Web Services. In addition to normal html-based Web interfaces, Google[8] and Amazon[9] recently published their Web Service interfaces (using SOAP and WSDL). Note that a machine cannot invoke or use normal html interfaces, whereas it can consume Web Service interfaces. Although their Web Services are not exactly the same as their html-based services, they published some of the key functions as a full-fledged Web Service.

For example, Amazon provides product catalogue Web Service and Google publishes spell-checking Web Service. Html versions of those services appear in Figure 9. In the demonstration, the general contractor agent (GCA) integrates these two Web Services. If we type an incorrectly spelled word such as "ELBIS PRESSLEY," Google will suggest a correct word "ELVIS PRESLEY," whereas Amazon can not search relevant catalogue information due to the misspelled word. Now, using their published Web Service interfaces, the GCA integrates two services as follows:

- First, consume Google's spell checking service first an incorrectly spelled word

- Then invoke Amazon's catalogue service with the returned correct word from Google.

Figure 10 shows that this integration process returned detailed inventory information from Amazon.

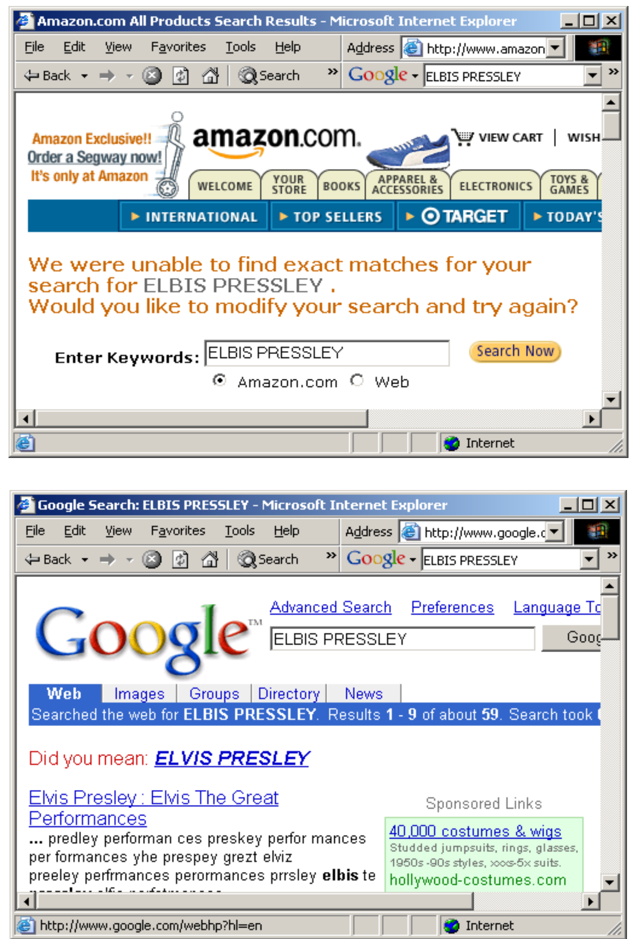


Figure 9: Different Search Results given a Wrong-Spelled Keyword

[8] http://www.google.com/apis/index.html

[9] http://www.amazon.com/gp/browse.html/103-2073630-6132643?node=3435361

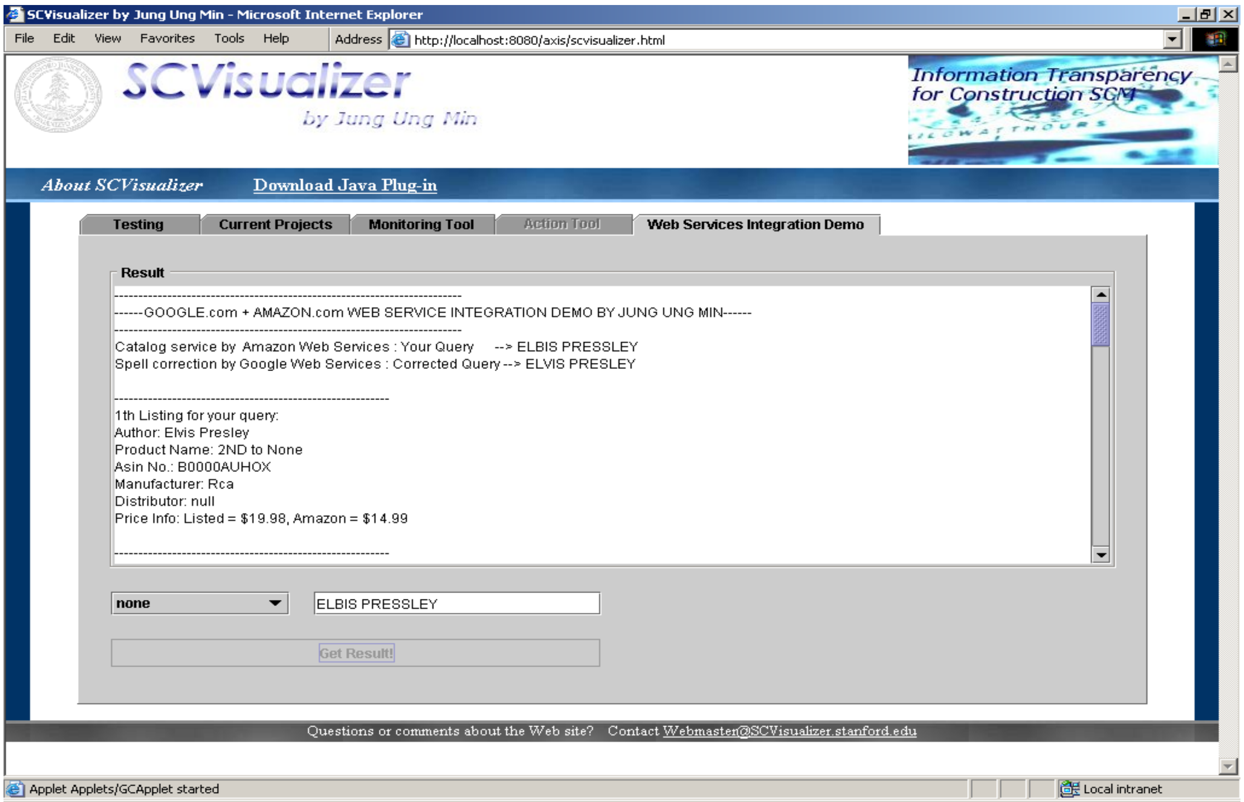


Figure 10: Demonstration of Integrating Non-agent Web Services

6. CONCLUSION

This paper presented *SCVisualizer*, a prototype system that facilitates real-time information sharing in construction environment. When the standards of Web Services were developed, specific target industries or specific application areas were not considered at all. Instead, they were designed for general-purpose usage. Decisions on detailed implementation methods and messages of services were turned over to application developers. This paper described how the existing Web Services standards could be leveraged in automating expediting process with computer agents.

To overcome the complexity and the redundancy of existing approaches, this paper proposed the AWSDL specification and independently distributed network framework as a prerequisite for automating the expediting process. Based on these frameworks, *SCVisualizer* was developed to facilitate real-time information sharing in construction supply chain management. *SCVisualizer* builds upon several components, such as structured data source, reasoning engine, and knowledge base combined with expediting interaction protocol. The demonstration shows that the proposed independently distributed framework and *SCVisualizer* can integrate various types of Web Services seamlessly, including non-agent based Web Services.

References

Agapiou, A., Clausen, L. E., Norman, G., and Flanagan, R. (1998). "The role of logistics in the materials flow control process." *Construction Management and Economics*, 16(2), 131-137.

Ambroszkiewicz, S. "Web service integration as a new paradigm for networked computing." *Proceedings of International Conference on Parallel Computing in Electrical Engineering*, 239-245.

Arbulu, R. J., and Tommelein, I. D. "Value Stream Analysis of Construction Supply Chains: Case Study on Pipe Supports Used in Power Plants." *10th Annual Conference of the International Group for Lean Construction*, Gramado, Brazil.

Ballard, G., and Howell, G. "What kind of production is construction?" *6th Annual Conference of the International Group for Lean Construction*, Guaruja, Brazil.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). "The Semantic Web." *Scientific American*(May).

Bogus, S., Songer, A. D., and Diekmann, J. "Design-Led Lean." *8th Annual Conference of the International Group for Lean Construction*, Brighton, UK.

Castro-Leon, E. (2002). "A perspective on Web Services."

Choo, H. J., and Tommelein, I. D. "WORKMOVEPLAN: Database for Distributed Planning and Coordination." *8th Annual Conference of the International Group for Lean Construction*, Brighton, UK.

Chua, D. K. H., Jun, S. L., and Hwee, B. S. "Integrated Production Scheduler for Construction Look-Ahead Planning." *7th Annual Conference of the International Group for Lean Construction*, Berkeley, USA.

Cooper, M., Lambert, D., and Pagh, J. (1997). "Supply Chain Management: More Than a New Name for Logistics." *The International Journal of Logistics Management*, 8(1), 1-14.

Howell, G. "What is Lean Construction - 1999." *Seventh Annual Conference of the International Group for Lean Construction*, Berkeley, CA, 1-10.

Huhns, M. N. (2002). "Agents as Web services." *Internet Computing, IEEE*, 6(4), 93-95.

Koskela, L. (1992). "Application of the New Production Philosophy to Construction." *CIFE Technical Report 72*, CIFE, Stanford University, Stanford, CA.

Kumar, K. (2001). "Technology for supporting supply chain management." *Communications of the ACM*, 44(6), 58-61.

Lane, R., and Woodman, G. ""Wicked Problems, Righteous Solutions" Back to the Future on Large Complex Projects." *8th Annual Conference of the International Group for Lean Construction*, Brighton, UK.

Lee, H. L., and Whang, S. (2000). "Information Sharing in a Supply Chain." *International Journal of Manufacturing Technology and Management*, 1(1), 79-93.

Lewis, I., and Talalayevsky, A. (1997). "Logistics and Information Technology: A Coordination Perspective." *Journal of Business Logistics*, 18(1), 141-157.

Martin, D., Burstein, M., Lassila, O., Paoluci, M., Payne, T., and Mcllraith, S. (2003). "Describing Web Services using DAML-S and WSDL."

McIlraith, S. A., Son, T. C., and Zeng, H. (2001). "Semantic Web Services." *IEEE Intelligent Systems*, 16(2), 46-53.

Nicolini, D., Holti, R., and Smalley, M. (2001). "Integrating Project Activities: the Theory and Practice of Managing the Supply Chain through

Clusters." *Construction Management and Economics*, 19(1), 37-47.

O'Brien, W., Issa, R., Hammer, J., Schmalz, M., Geunes, J., and Bai, S. (2002). "SEEK: Accomplishing Enterprise Information Integration Across Heterogeneous Sources." *ITCON-Electronic Journal of Information Technology in Construction, Special Edition on Knowledge Management*, Vol. 7, 101-124.

Petrie, C. B., C. (2003). "Service agents and virtual enterprises: a survey." *Internet Computing, IEEE*, 7(4), 68-78.

Preist, C., Byde, A., Bartolini, C., and Piccinelli, G. (2001). "Towards Agent-based Service Composition through Negotiation in Multiple Auctions." *HPL-2001-71*, Hewlett-Packard Company.

Simon, S. J. (2001). "The Art Of Military Logistics." *Communications of the ACM*, 44(6), 62-66.

Taylor, J., and Bjornsson, H. C. "Identification and Classification of Value Drivers for a New Production Homebuilding Supply Chain." *10th Annual Conference of the International Group for Lean Construction*, Gramado, Brazil.

Tosic, V., Mennie, D., and Pagurek, B. (2001). *Dynamic Service Composition and Its Applicability to E-Business Software Systems*, Budapest, Hungary.

Vaidyanathan, K. "Case Study in Application of Project Scheduling System For Construction Supply Chain Management." *10th Annual Conference of the International Group for Lean Construction*, Gramado, Brazil.

Vrijhoef, R., and Koskela, L. "Roles of Supply Chain Management in Construction." *7th Annual Conference of the International Group for Lean Construction*, Berkeley, USA, 133-146.