ENGINEERING SIMULATIONS WITH WEB-BASED SERVICES

Jinxing Cheng¹, Chin Pang Cheng², Mai Anh Le Thi³ and Kincho H. Law⁴

ABSTRACT

The web services model is becoming a popular approach for integrating software applications to improve the flexibility and extend the functionalities of a software application by making it interoperable with other application services. With the continuing rapid growth of online information sources, the integration of such services can further include many up-to-date information for engineering simulations. Three example applications are presented to demonstrate that the web service approach can potentially have significant impacts in facility design and engineering.

KEY WORDS

web services, engineering simulation, project management, design, procurement

INTRODUCTION

As computer programs become ever more complex, the trend of software development has shifted from focusing on programming towards focusing on integration, as illustrated in Figure 1 (Beringer et al. 1998). This trend is of no exception in the A/E/C industry. As stand-alone engineering applications (such as structural analysis, CAD, project management, etc..) of the 1960s and 1970s began to mature, the industry has been pushing for shared or integrated platforms, extending domain-specific tools and capabilities beyond an individual application to support project-wide activities (planning, design, construction, business transactions and other functions). In parallel to this trend, as communication technologies advance, there is a shift from stand-alone desktop applications toward distributed (e.g. Webbased or Web-enabled) services. Many new developments and business models will take advantage of advanced communication technologies and will be based more and more on the composition and integration of existing application components.

With the rapid development of the Internet and networking technologies, the computing environment is evolving toward an interconnected web of autonomous services, both inside and outside of enterprise boundaries. A "web service" can be described as a specific function

¹ Formerly PhD Student, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA, 94305, USA, jim.cheng@oracle.com

² PhD Student, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA, USA, <u>cpcheng@stanford.edu</u>

³ Research Engineer, Statsbygg, the Directorate of Public Construction and Property, P.O. Box 8106 DEP., N-0032, Oslo, Norway, <u>malt@statsbygg.no</u>

⁴ Professor, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA, USA, <u>law@stanford.edu</u>



Figure 1: Trend of Software

that is delivered over the Internet to provide a service or information to users. Users of information systems now have tremendous sources of services on the Web: they can manage a bank account, purchase books, arrange travel reservations, and track shipping packages. Web service integration is important for the automation of applicationto-application or organization-to-organization cooperation using the Internet infrastructure. However, it is not easy for a user to perform a complex task which composes of many

sub-tasks and requires access to many web services.

The web services model has become a favorite approach for integrating software applications in that the model can improve the flexibility and extend the functionalities of an application by making it interoperable with other software services. An engineering simulation may now involve a number of geographically distributed software applications. Additionally, online information can be dynamically integrated with the applications. The simplicity of the web services model makes it possible to build a complex software system incrementally. In this paper, we describe a few examples to illustrate the potential applications of web services, ranging from project management to design and procurement services.

A FRAMEWORK FOR INTEGRATION OF WEB SERVICES

With the emergence of Web-based software applications, many languages have been proposed to facilitate the reuse of Web services or software components. Examples include Web Services 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). These service description languages, however, are mainly targeted for business oriented applications and are not designed for managing and reusing information to support engineering applications. In the engineering and construction domain, many stand-alone applications (e.g., Microsoft Project, Microsoft Excel, Primavera Project Planner, and AutoCAD) are widely used. These tools are designed for specific applications. Even with many existing online Web services, such as weather forecasting, product catalogues, finance reports, etc., these services are not readily integrated with traditional standalone applications.

For web service applications, software components exist as autonomous services managed under multiple administrative domains and controlled by different service providers. The loosely coupled, distributed web services model allows new services to be added as needed during runtime. Our prototype web service simulation framework builds upon a software composition infrastructure based on data and control flows (Liu et al. 2002) and software wrappers. The Flow-based Infrastructure for Composing Autonomous Services (FICAS) (Liu 2003) is utilized to invoke distributed services and to direct data flow among

different services. While there are other solutions available for distributed service invocation, including Remote Procedure Call (RPC), Common Object Request Broker (CORBA), and Simple Object Access Protocol (SOAP), FICAS is designed to handle applications that involve high volume of data typically found in engineering applications. Specifically, FICAS takes advantage of distributed data flows to efficiently route the data to designated applications. Wrappers are developed with information exchange protocols based on Extensible Markup Language (XML) and the Process Specification Language (PSL) (Cheng et al. 2003).

In order to describe the usage of web services, a simple, easy-to-use simulation access language, SimAL, is designed and implemented to coordinate application tools and to simulate scenarios in assisting decision making (Cheng 2004). In general, three key factors are involved in decision-making: alternatives, information, and preferences. Alternatives imply that more than one option should be available. Information refers to the knowledge available to users about different options. Preferences specify the aspects that users want to optimize. To support these functions, operational statements for Invocation, Operation, Control, and Decision-Support are provided in SimAL (Cheng 2004).

The objective of our research in web services is to develop methodologies that can effectively wrap legacy applications and make them accessible over the network. Different applications would require different models for integration and coordination. Detailed discussion of the simulation framework and the integration architecture is beyond the scope of this paper. Interested readers are referred to the works by Liu (2003) and Cheng (2004) for details. Instead, a number of demonstrative examples are employed here to illustrate the methodologies and potential applications of engineering web services.

APPLICATION EXAMPLES

To illustrate the web-based service integration framework, commercial software applications, such as Microsoft Project, Microsoft Excel, Primavera Project Planner, Vite SimVision, Autodesk Architectural Desktop (ADT) and 4D Viewer are wrapped as web services that export their functionalities (see Figure 2). Applications run on heterogeneous platforms can be accessed homogeneously through standard web services interfaces. Functionalities from various software applications can be brought together to complete a specific engineering

The prototype also task. incorporates a variety of devices ranging from PDA. web browsers, desktop computers, and computers server to support ubiquitous access to the information and simulation applications. Additionally, online services, such as weather forecasting and product information. also are



Figure 2: Demonstration Infrastructure for Web Services

included. In short, by using the web services model to develop the integration framework, engineering applications can interoperate regardless of locations and platforms.

EXAMPLE 1: INTEGRATION OF CAD AND SCHEDULING TOOLS

CAD and scheduling tools are among the most important software applications in design and construction. This example illustrates how to integrate CAD and scheduling tools using the simulation framework. To build the link between a CAD model and a schedule, the 3D model needs to be object based. In addition, objects in CAD models should be grouped and associated with corresponding tasks in the project schedule. Once the association is established, the program can display the 3D model according to a specified schedule. For each object (or a set of objects) in ADT, the program first retrieves the corresponding scheduling task from the association table. It then compares the targeted display date against the start and finish dates of the task to determine whether the object should be displayed, partially displayed, or not displayed at all. The appropriate CAD models are created to show the construction progress on the targeted date. Using a Web browser, users can view the CAD model of the project at various dates without the need of having ADT installed on their computers. This functionality can be very useful to field personnel who may not always have access to computers that host complex CAD tools; on the other hand, it is not unusual for them to have PDA or laptop computers, as well as Internet connections. It should be pointed out that the CAD models are dynamically generated by ADT and Primavera P3 according to the latest model and scheduling information. This approach is different from Web-based project repositories, which do not offer real-time analysis tools to process project information. Once the connection between the CAD model and the schedule, users can adjust task schedules, update CAD models and view the results using a Web browser. This process is depicted as shown in Figure 3 and can be implemented as a program written in the SimAL language.

To illustrate, as shown in Figure 4, suppose that the user modifies the duration of task *ID5* ("Grading/Excavation") from 55 to 85 days and update the schedule on a Web browser. Once the simulation program receives the change, it notifies Primavera to reschedule the



Figure 3: Process for Integrating Project Schedule and CAD Model Display

project (see Figure 5). When rescheduling is completed, the simulation program proceeds to instruct ADT to incorporate updated scheduling information and to display the CAD models corresponding to the updated schedule, as shown in Figure 6. The changes in the construction progress can also be viewed on a Web browser, as shown in Figure 7.

AEC DCS Dem	onstration - SCHEDI	ULE - Microsoft Internet Explo	orer		AEC DCS Demo	onstration - SCHEDI	JLE - Microsoft Internet Explorer		
File Edit View	v Favorites Tools	Help			File Edit View	Favorites Tools	Help		
$ \Rightarrow Back \ \bullet \ \Rightarrow$	· 🔘 🖻 🕼 🔘	Search 🔂 Favorites 🎯 Medi	6 🕑 💁 🎯 🖬 .		↓ Back + →	0 2 4 0,	Search 🕞 Favorites 🎯 Media 🎯	B- 4 00 1	
Address 👸 http:	//psl.starford.edu:808	80/DC/servlet/DCS?ProjectId=MCE	DO&TableName=SCHEDUU	8.TimeStamp=02-20	0-21 Address 🗃 http:,	//psl.stanford.edu:808	0/DC/servlet/DCS?TableName=SCHEDL	.E&TimeStamp=02-3	0-2004+15%3A01%34
SCHEDULED	STARTDATE	DURATION	FREEFLOAT	TOTALFLOAT	SCHEDULEID	STARTDATE	DURATION	FREEFLOAT	TOTALFLOAT
ID0	06-21-2002 00:00:00	0	Update 0	0	ID0	06-21-2002 00:00:00	0 Upde	0	0
ID1	06-21-2002 00:00:00	45 0	Update 355	355	ID1	06-21-2002 00:00:00	45 Upde	355	355
ID2	06-21-2002 00:00:00	74 <u> </u>	Update 326	326	ID2	06-21-2002 00:00:00	74 Upde	326	326
ID3	06-21-2002 00:00:00	6	Update 394	394	ID3	06-21-2002 00:00:00	6 Upde	394	394
104	06-21-2002 00:00:00	[35	Update 0	•		06-21-2002 00:00:00	35 Upde	0	0
IDS	08-09-2002 00:00:00	55 <u> </u>	Update 0	0	пз	08-09-2002 00.00.00	85 Upde	0 0	0
ID7	08-09-2002 00:00:00	22	Update 0	33	ID7	08-09-2002 00:00:00	22 Upde	0	33
ID8	09-10-2002 00:00:00	17 U	Update 326	326	ID8	09-10-2002 00:00:00	17 Upde	326	326
ID6	10-25-2002 00:00:00	75 U	Update 0	0	1D6	12-06-2002 00:00:00	75 Upde	0	0

Figure 4: Modifying Schedule on a Web Browser

			3 3					2
Description	Dur Start	J J A S O B D J F M A M F J A S O B	8 3 8 1	B	Description	Dar Start	PERSONAL DEPENDENCE NO. 1 PERSON NO. 1	TTIN
					A CONTRACTOR OF A CONTRACTOR O			
Slat	0 21.JUN02	Start	1 (E E	60	IRM1	0 21,0402	Terration and the second	
Demotion	35 21.A.M02	Demolition		6.4	Denotion	30 21,0902	Annual Demolition	1 1
Pernito	74 21 JUN02	4	T Permi	02	PHINES	74 2130402		
Mobilization	45 21.8,N02		V Mobili	DI	Mobilization	45 21,0962		-
Tree Relocation/Rend-16	6 21.A.NO2		W Tree F	613	Thee Relocation Planckie	6 2130402		
OradingExcavation	55 09AU002	Amount of Grading Escavation		6/5	Creange:cevator	85 (39AU002	Accession Grading Exception	
Underground Utilities	22 09AU002	C		07	Underground Utilities	22 (9MU002	Coulor ground Utilities	
Hardscape Ste	17 1058902		-		Plandscape Site	17 105EP02		
Peting Structure	75 250CT02	Parking Structure			Perking Structure	753086€C02	Parking Structure	
Landocaping	46 250CT02		T Landson	09	Landscaping	46 0808C02	A	- 71
Superstructure Franking	130 07978003	The second	Contractory 1	Ott	Superstructure Friending	130 21MARUS	Alexandra and a second and a second and a second	/ranning
Code nated	132 0777003		IT CARD	010	SVF Fotel	132 21MARD		
There are a second seco	112 0041003		The second second	014	Perrices	112 1958900		
Provide Providence	112 00-0000			O13	Poor a stan	06 19(EPG)		- 91
	10 000400003		1	012	Extense Tren & Trenks	10 1958903		
Elsenor min a mess	10 06A0003		Esterior	D15	Punch & Obje Out, Building	7 030C703		
Punch & close out, Building	7 [22A0003		Punch	D16	Punch & Close Out, Silework	7 24FEB04		-
IPUPUD & CAME ONE SERVICE	7113JAND4		Panch	017	Freish.	D (GAMAROA		

Figure 5: Re-scheduling Project Activities using Primavera P3



Figure 6: Displaying 3-D Models According to Schedule Changes on AutoCAD ADT

Joint International Conference on Computing and Decision Making in Civil and Building Engineering June 14-16, 2006 - Montréal, Canada



Figure 7: Visualizing Model Change on a Web Browser

EXAMPLE 2: INCORPORATING ONLINE WEATHER INFORMATION FOR SIMULATION

The purpose of this scenario example is to illustrate bringing on-line services to engineering simulation. Figure 8 shows an example workflow to include weather conditions in project management. A parser is developed to convert the weather forecast service information into XML format as shown in Figure 9. In addition, as depicted in Figure 10, a simulation program written using SimAL is embedded in Microsoft Excel to specify the workflow. Figures 11 and 12 illustrate the impacts of the weather conditions to the schedules displayed in Primavera P3 and the results of task and resource backlogs generated by Vite SimVision and displayed as charts using Microsoft Excel.







Figure 9: Expressing Weather Information in XML

Joint International Conference on Computing and Decision Making in Civil and Building Engineering June 14-16, 2006 - Montréal, Canada

the fift year frant P	grmat Icols D	eta Window Help	x A 21 21 10 43 (2) 1 Prove	{ /* Establish Connections */
134	•	0	P P	() Establish Connections ()
Project Name	MCDO	0		$p_3 syc = SETUP("ServiceP3")$
Location	West Palm I	Jeach, FL		po_site shirter(services)
Zip Code	33410			nsl_svc - SETUP("ServicePsl")
Services	Microsoft Pr	siect		psi_sve = bETOT(betviee sr)
	Vite SimVisi	on		vite svc - SETUP("ServiceVite")
Elimitate Browner	Weather Sei	vice	and the second se	vite_sve = SETOT (Service vite)
Simulate Program	C Enter the	simulation code in t	se textbox:	notification svc = SETUP("ServiceNotification")
	SimAL We	atherDerno		
	1 10	_evc = SETUP("Serv	ceP3")	where $cast svc = SETUP("Service WeatherForecast")$
	po	Levc = SETUP("Ser	icePsi")	
	no	tification_svc = SET	IP("ServiceNotification")	wprocess $svc = SETUP("ServiceWeatherProcess")$
	w	orecast_svc = SETU	("ServiceWeatherForecast")	
		HOCARE AND - SELIO	(Datate available to case)	\rightarrow /* Invoke Services */
		I = psl_svc.INVORE('to-psi", %%)	/ myoke bervices /
		= wprocess_svc.lk?	ORE("ProcessForecast", wf.an	2 nsl = nsl syc INVOKE("to-nsl" %%)
	atho, %%%)			
	9.3	e = vite_svc.INVOKE	"simulate", arho1, %%)	V = wforecast syc INVOKE("RetrieveForecast" %%)
				wi = wiorceast_sve.invvorte(inclusit , /0/0)
	1	of - notification svc	WYORET bil standord adu", 8250.	wp = wprocess svc.INVOKE("ProcessForecast", wf arho, arho, %"
	e Sale	ect Simulation File	C.\Develop\SinvAL\P3Arg_10.txt	$r_{2}^{2} = r_{2}^{2}$ give N UOKE("reschedule", yw orbe $0(0/)$
				$p_3 = p_3$ svc.invOKE(reschedule, wp_arno, %%)
	_	Dimutete		vite = vite_svc INVOKE("simulate"_arho1_%%)
				the = the_stell() shall be another ()
				notif – notification syc INVOKE("nsl stanford edu" 8250 status)



	navera Projecc Plan	iner -	- [Manual							navera Project Plan	ner -	[Manual				
🚰 File	Edit View Insert	For	rmat Tools	Data Wind	dow	Help			🚰 File	Edit View Insert	For	nat Tools	Data Wind	ow Help		
		*	B (**		9	N 1990-	= 🛻 🖪 🗶 🛤 📖 🗖	7 🚣 🗈 🖂			*		<u></u> # #	🕙 😼 🗉	0-3 ABX = = = = 7	
-+	*								-+							
Activity	Activity	Orig	Early	Early			2003		Activity	Activity	Orig	Early	Early		2003	
10	Description	Dur	Start	Finish	E.	1 # 15 22 21	OCT NOV DEC	22 29 5 12 19 26 J	10	Description	Dur	Start	Finish	A SEP	OCT NOV DEC	JAN FE
1D100	Stort	1	1 01SEP03	01SEP03	14	Start	1 1		ID100	Stort	1	01SEP03	01SEP03	T Start	: :	:
D110	Grade Lot	4	4 025EP03	055EP03	13	Grade Lot			ID110	Grade Lot	4	025EP03	05SEP03	Grade L	ot	
ID120	Ley Foundation	26	5 00SEP03	100CT03			Ley Foundation		ID120	Ley Foundation	30	085EP03	170CT03		Ley Foundation	
ID130	Foundation Complete	0	0 130CT03	100CT03		_	Toundation Complete		ID130	Foundation Complete	0	200CT03	17OCT03	11 -	Troundation Complete	
ID140	Frame House	8	9 13OCT03	23OCT03			Frame House		D140	Frame House	9	200CT03	300CT03	11	Frame House	
ID170	Install Plumbing	10	0 240CT03	06NOV03			Install Plumbing		ID170	Install Plumbing	10	31OCT03	13NOV03		Install Plumbing	
ID180	Frame Roof	10	0 24OCT03	06NOV03			Frame Roof		ID180	Frame Roof	10	31OCT03	13NOV03		Frame Roof	
ID160	Install Electrical	3	3 24OCT03	28OCT03			AND Install Electrical		ID160	Install Electrical	3	31 OCT03	04NOV03		An Install Electrical	
ID150	Install Heating	2	2 240CT03	27OCT03			A Install Heating		ID150	Install Heating	2	31OCT03	03NOV03		📩 🗸 🗸 Install Heating	
ID190	Ready For Drywall	0	07NOV03	06NO/03			Ready For Drywall		ID190	Ready For Drywall	0	14NOV03	13NOV03		Ready For Drywall	
ID200	Instal Drywal	26	8 07NOV03	16DEC03			A	Install Drywall	ID200	Install Drywall	20	14NOV03	23DEC03			stall Drywall
ID210	Paint	0	8 17DEC03	26DEC03				V Paint	ID210	Paint	8	24DEC03	02JAN04		A=	V Paint
10220	Install Appliances	2	2 29DEC03	30DEC03				🌌 Install Applianc	ID220	Install Appliances	2	05JAN04	05JAN04			Install Appliances
ID230	Finish	0	31DEC03	30DEC03				I Finish	ID230	Finish	0	07JAN04	06JAN04			Tinish 🔤
· ·															- ·	

Figure 11: The Impact of Weather on the Schedule Displayed in Primavera P3



Figure 12: The Impact of Weather on Task Backlog Displayed as Charts in Excel

EXAMPLE 3: INCORPORATING ONLINE CATALOGUE INFORMATION FOR DESIGN

With the rapid increase of online catalogs, the interactions and the information flow between product descriptions and design activities can be valuable. Manufacturers can publish their product information and professional designers can acquire design data online. One example is Autodesk's i-drop technology (see http://www.autodesk.com/idrop) which enables a web developer to add drag-and-drop capabilities to a web site, which then acts as an open gateway between the web content and the web user. As shown in Figure 13, the content of a product is expressed in HTML and the product information can be structured in an XML file, which can also includes other files (such as drawing or image) to be transferred. Designers can easily incorporate manufacturing products into their design. Product information such as model

code, manufacturer, supplier, price quote, etc. can be attached as object attributes. Furthermore, once the object is dragged-and-dropped into the design drawing the designers can update the information to produce a procurement list during the design phase.

Take a floor plan of a residential building shown in Figure 14 as an example. The designer can drag items found on different web catalogues into the design drawing. The inherited information about the items are also included. Once the items are added into the design, the designer can copy and paste the downloaded objects as desired. The designer can also change the values of the attributes. Figure 14 shows the floor plan with furniture and utilities imported into the design. Figure 15 shows a spreadsheet tabulating the items and their descriptions. The data can be outputted in XML format and forwarded to other applications.

A "virtual" supply chain is a vital element of today's business environment. Companies around the world are trying to take advantage of the Internet and information technologies to create virtual supply chains where customers, suppliers, and business partners collaborate with each other. Our current research is to utilize FICAS and SimAL to create a demonstration prototype modeling a procurement supply chain as shown in Figure 16. An organization holds information about corresponding entities and keeps monitoring key data, such as inventory level, order and delivery status. Project participants can gather information with schedule data so that they can forecast potential problems. If problem arises, others across the chain can quickly be informed so that new services and strategies can be deployed to resolve issues at hand.





Joint International Conference on Computing and Decision Making in Civil and Building Engineering June 14-16, 2006 - Montréal, Canada



Figure 14: Drag-and-Drop Online Product to a Design Drawing

arial		- 10 -	R Z II		a a	e l che che l						
	B27 -	A Di	shwasher			o se se						
	A	B	c	D	E	F	G	н	1	J	K	L
1			Currer	nt Inve	ntory Re	quired	Upd	ated on	2/10/200	6 16:39		
2	Product Code	Product	Manufacturer	Trade Name	Model Number	Material	Supplier	Price	Color	Quantity		Total # of
3	K-CA-0789	Bed	Carpenter		SB4422TOWT	Wooden	IKEA	\$599.00	White	2	\$1,198.00	
4 1	K-CA-9034	Chair	Carpenter	CIFE		Wooden	IKEA	\$69.99	Khaki	7	\$489.93	Total C
5 1	HD-CA-0029	Table	Carpenter	Patricia	12193-69	Wooden	HomeDepot	\$499.00	Dark Brown	1	\$499.00	l otal C
5 1	K-UN-3482	Table		Tresidder	MOK-9	Plastic	IKEA	\$99.99	Red	1	\$99.99	
1	HD-SO-7872	Sofa	Softy	Tri-soft		Leather	HomeDepot	\$2,399.99	Black	1	\$2,399.99	
	HD-SO-9923	Sofa	Softy	Single-soft		Leather	HomeDepot	\$899.99	Black	1	\$899.99	
ī,	HD-SO-8973	Love Seat	Softy	Single-com	i i	Leather	HomeDepot	\$1,299.99	Red	1	\$1,299.99	
	K-CA-8723	Table	Carpenter	Patricia	12188-0A	Wooden	IKEA	\$149.99	White	1	\$149.99	
i li	K-UN-0093	Lamp			P03797989	Metal	IKEA	\$129.99	Silvery	1	\$129.99	
2 1	K-UN-0020	Sink		UNAKO	E121K	Metal	IKEA	\$100.00	Silvery	1	\$100.00	
8 8	HK-MO-7890	lavatory	Mohla	Comfort	8937HK	Acrylic	HomeDepot	\$340.00	Dark Blue	3	\$1,020.00	
	HK-MO-1121	toilet	Mohla		WM-654	Acrylic	HomeDepot	\$279.00	White	3	\$837.00	
5	HD-MO-2218	bath	Mohla		RTA-23	Acrylic	HomeDepot	\$888.00	Dark Blue	2	\$1,776.00	
1	HK-MO-6633	Grab Bar	Mohla	Franken	HYAM334	Metal	HomeDepot	\$20.00	Silvery	1	\$20.00	
	HK-MO-6630	Grab Bar	Mohla	Franken	HYAM280	Metal	HomeDepot	\$59.99	Silvery	1	\$59.99	
8 1	K-CA-9932	Table	Carpenter	Jackson	TY99U	Wooden	IKEA	\$259.99	Brown	1	\$259.99	
9 1	K-CA-3434	Piano Chair	Carpenter	Jackson	TX38A	Wooden	IKEA	\$350.00	Black	1	\$350.00	
	HD-TP-3343	Refrigerator	Trapot	Tech	KHLE8097202	Metal	HomeDepot	\$538.00	White	1	\$538.00	
1	HD-TP-7784	Washer	Trapot		HODHL2983	Metal	HomeDepot	\$599.99	White	1	\$599.99	
2	HD-TP-24578	Dryer	Trapot		WAHDM77682	Metal	HomeDepot	\$369.00	White	1	\$369.00	
3 1	K-SA-1200	Stove	Samo	Safe	KK-903	Metal	IKEA	\$1,599.99	White	1	\$1,599.99	
	K-0S-7872	Desk	OfficeSmart		AKKEM389	Wooden	IKEA.	\$230.00		1	\$230.00	
5	HD-PO-2304	Television	Polar	Crystal	JJX48204H	Plastic	HomeDepot	\$1,399.99	Black	1	\$1,399.99	
5 1	K-0S-3388	Bookshelf	OfficeSmart		BKSEM887	Wooden	IKEA	\$150.00		1	\$150.00	
1	K-SA-8973	Dishwasher	Samo	Speed	LL-802-9B	Metal	IKEA	\$400.00	White	1	\$400.00	
1Ē			T									

Figure 15: A Spreadsheet Enlisting Items Inserted in the Design



Figure 16: An Illustration of Procurement Supply Chain

SUMMARY AND DISCUSSION

This paper has presented three example applications that demonstrate the potential applicability and flexibility of the web services technology. Integrating distributed

engineering applications as web services provides an effective mechanism to extend the functionalities of legacy applications and make them more accessible to a broader group of users. With the flexibility and scalibility of the web services technology, the impact could have significant impact in project life cycle design, operations and management.

Web service is still an emerging technology, and many improvements need to be made. Many general-purpose features, such as security, reliable message delivery, and transactional semantics, are needed to facilitate the development of web services. Specifications for the web services need to be standardized for handling heterogeneous data formats and software platforms. To promote scalability in the integration of web services, programming models need to shift from procedural call style services toward specification-centric services. As these new developments progress, the web services technology will be applied more widely for developing engineering software applications.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Chuck Han for his assistance on the i-drop technology. Mr. Xiaoshan Pan provided the original floor plan shown in Figure 14. The authors would also like to acknowledge the software support from Autodesk Inc..

REFERENCES

- Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D. Thatte, S. Trickovic, I., and Weerawarana, S. (2003), *Specification: Business Process Execution Language for Web Services (BPEL4WS)*, Version 1.1., http://www-106.ibm.com/developerworks/library/ws-bpel/.
- Ankolekar A, Burstein M, Hobbs JR, Lassila O, Martin DL, McIlraith SA, et al. (2001), "DAML-S: Semantic Markup for Web services," *Proceedings of the International Semantic Web Working Symposium*, Stanford, CA, pp. 411-430.
- Beringer, D., Tornabene, C., Jain, P., and Wiederhold, G. (1998), "A Language and System for Composing Autonomous, Heterogeneous and Distributed Megamodules," *DEXA International Workshop on Large-Scale Software Composition*, Vienna, Austria.
- Cheng, J., Gruninger, M., Sriram, R.D., and Law, K.H. (2003), "Process Specification Language for Project Scheduling Information Exchange," *International Journal of IT in Architecture, Engineering and Construction*, 4:307-328.
- 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.
- Leymann, F. (2001), *Web Services Flow Language (WSFL)*, Version 1.0, IBM Corporation, <u>http://www-306.ibm.com/software/solutions/webservices/pdf/WSFL.pdf</u>.
- Liu, D., Law, K. H., and Wiederhold, G. (2002), "Data-flow Distribution in FICAS Service Composition Infrastructure," *Proceedings of the 15th International Conference on Parallel and Distributed Computing Systems*, Louisville, KY., 2002.
- Liu, D. (2003), A Distributed Data Flow Model for Composing Software Services, PhD Thesis, Department of Electrical Engineering, Stanford University, Stanford, CA.