Stylus Systems Pvt. Ltd.

7, Park Avenue
Horamavu Agara
Bangalore, India 560 043

Phone +91 80 545 5070
Email : info@stylusinc.com
Url : www.stylusinc.com

Contents

Chapter 1

The Web Services World

Introduction

With the web came the worldwide library of information. The web in its first avatar provided a window for anyone to locate / exchange information. The users experience was enhanced with the appealing and valuable information that was presented. But with all its charms the web continued to remain just an online lobby for most companies. The promises of interoperability for applications with standards like CORBA and COM and its avatars failed to deliver.

With web services the web seems to be maturing and finally delivering its promise of a true backbone for applications to interact and transact. The web in its second generation promises to be a network of applications optimized to serve the users needs. Web Services are the first step in this direction.

The web in its second generation promises to be a network of applications optimized to serve the users needs.

What is a web service?

Definition

Web Services are business process interfaces that are based on open Internet standards. They use standard protocols like WSDL to describe themselves, UDDI to advertise and syndicate and SOAP to communicate. They are created and exposed as parts (hooks) of a bigger Internet application to utilize key services so that they can either provide or require collaborative other services (parts of other internet applications). In other words, with Web services an application can expose its public functions for collaboration while continuing to keep it secure. And this can be accomplished using open standards so that newer applications can always collaborate and build upon the existing ones.

Web Services are business process interfaces that are based on open Internet standards.

Some Examples

Imagine having to go on a holiday from New York to Bangalore. If you need to make your bookings online the first step would be to contact a flight information site and make a suitable ticket booking. Then you would have to make your accommodation arrangements by locating appropriate hotels and resorts in Bangalore. Today, in the present web scenario, this process would turn out to be labor-intensive and require a great deal of intervention from your side. But with web services you could specify your destination, the budget, the dates and even your preferences in cuisine and the application would take care of locating the correct flight, hotel within budget and preferences without any manual intervention and also confirm your transaction in an instant.

Imagine having your PDA showing your current stock rates online and also helping you to make your investments minute to minute. With web services your PDA stock application could automatically query a stock exchange web service at regular intervals and bring you the info you need, and help you make the transactions too.

Imagine your futuristic refrigerator using the grocery store web services, across the home LAN over the Internet, to order more apples when your fruits tray goes empty.

Where are they used?

"Web services are a very profound advancement for e-business. [They] can change the Internet into a platform for application integration," says Scott Hebner, IBM's director of marketing for WebSphere.

Early examples of Web services include on-demand delivery of stock quotes to a cell phone or pager -- but such bare-bones capabilities are merely the tip of the iceberg.

Web services encompass just about any application available today on the web or for that matter any application that will be delivered in the future. Web services use the standard protocols like UDDI (Universal Description, Discovery and Integration), XML (extensible Markup Language), SOAP (Simple Object Access Protocol), and WSDL (Web Services Description Language). This would mean an entirely new market place. A place where Service Providers publish their services and users discover and use these services, all in a common and easily exchangeable format.

The Service Providers World

The Service Providers World would revolve around enabling key services of the business processes of clients as web services that can the accessed by Users. This would mean creating web service interfaces to existing applications so that they can be made accessible across the web, maybe even integrating applications to provide Web services interfaces. It could even mean creating new applications that enable your key services as web services. The Services Provider then joins the fray of publishing his services amongst scores of competitors and the Internet transforms itself into an innovative market place.

The Users World

Users will no longer need to manually query and dig for services that they need. All regular applications will soon be web services enabled and will search and transact according to individual preferences. The choice is bigger and the bargains are better. The user goes to a common marketplace to locate the best Service Provider and transact with him based on user-specific preferences.

The New Market Place

With Users and the Services Providers coming to a common marketplace it will soon mirror our regular market places of today - the typical world of vendors, buyers and brokers. Yes, the brokers will be there too. The consolidators and integrators will become brokers who provide consolidated web services that in turn depend on other vendor web services.

A company offering a particular service describes it with WSDL and then registers it with a directory, such as UDDI. A user in need of that service can then look it up in the directory and access it using XML and SOAP. For instance, a company conducting international commerce can access a service that approves global tariffs without having to own the tariff application or even subscribe to it in an Application Service/Subscription fashion.

There is no question that if Web services are to take off as smoothly as vendors hope, a significant chunk of the more than 20 million programmers in the world will have to write to UDDI, WSDL, XML, and SOAP. "The promise of Web services depends heavily on vendors supporting the standards," IBM's Hebner says. Although analysts say that more standards are likely to develop, the four initial efforts will form the backbone of Web services. "These de facto standards appear to have momentum that [proves] they are going to be important," says Al Gillen, a research manager at market research firm IDC, in Framingham, Mass.

The most important benefit of these standards is that client applications based on products from one vendor will be able to communicate with Web services even if they are based on software from another vendor, according to Barry Goffe, Microsoft's .NET product manager. "Web services will interoperate seamlessly, but probably not until about the third version," Gillen adds. "The first version will probably have a few glitches."

Web services will change not only the nature of how applications are used, but also the way they are developed. "This is a radical shift for companies to expose their infrastructures to the Internet," says Shawn Willett, a principal analyst at Current Analysis, based in Sterling, Va. Although a few of the Web-services vendors are trying to persuade the public that the development mind-set requires only slight alterations, developers and analysts alike are skeptical, if not downright disbelieving.

A report published in late January by Stamford, Conn.-based consultancy Gartner predicts that all leading e-business platforms will support at least the basic Web-services infrastructure by 2002 or by 2003. By 2003, more than 75 percent of Web services in production will be supported by Web-services infrastructures provided by IBM, Microsoft, or two or three other vendors, the report states.

Why Should I use them?

Any company considering the move should ask themselves what their business needs are, what are the infrastructure needs those business objectives create, and how they plan to communicate with Internet exchanges, according to Muntuck Yap, a principal in the Internet services division at Arthur Andersen, in Chicago.

"If you don't think about what a Web service can be, and design for that, you'll only be able to take advantage of about 25 [percent] to 50 percent of the benefit of these technologies," says David Story, vice president of engineering at Allegis, a San Francisco-based PRM (partner relationship management) vendor that provides its software in the form of a Web service to Fortune 500 companies.

Benefits

Web services go beyond software components, because they can describe their own functionality, look for, and dynamically interact with other Web services.
Web services provide a means for different organizations to connect their

applications with one another and to conduct dynamic e-business across a network, no matter what their application, design or run-time environment. From a business standpoint, Web services offer a new range of possibilities for how organizations and their partners develop business solutions.

In the years of building business applications, you have had to know who was using the applications, how, when, where and for what. These were the boundaries within which you used to have to create and use applications. Now with Web services, you can build applications without having to know whom the users are, where they are, or anything else about them. And as a user of these applications, you can source them as easily as you would be able to source static data on the Web, with complete freedom and no concern about the format, platform, or anything else. You can just get it.

Even more revolutionary, Web services are self-integrating with other Web service applications. Until now, using traditional software tools to make two e-business technologies work together required lots of work and planning: you had to agree on the standards to pass data, the protocols, the platforms, etc. Now with Web services, applications written to the new standards will be able to automatically integrate with each other wherever they originate. So it will not require any work to integrate the applications -- it will happen automatically. This helps to makes e-business - dynamic e-business.

Myths

The first common myth is ***"With Web services I have to throw out all my old applications out of the window".*** In fact with web services you can re-use all your old applications by enhancing them as web services. This is one of the biggest advantages of recent times with technology. Re-using any of your old applications whether it is a legacy system, a client service application or a web application – they can all interoperate and work together and be bound together using the web services thread.

The second common myth is ***"I have to learn/use new technology / platforms for using web services".*** Web services can interoperate across technology, platforms and development languages. An ASP application can work with JSP and whether it is on the Linux platform or the Microsoft platform does not matter. They can work across the Internet and across applications.

Process Reengineering

As the marketplace evolves we will find web applications automatically being engineered to take advantage of the web services. This will stream line the business processes involved and make the Internet itself one large application that serves your business.

Chapter 2

Identifying my Web Services

What can be a web service?

Web services are self-contained, modular business process applications, which are based on open, Internet standards. Using the technologies of WSDL (to describe), UDDI (to advertise and syndicate), and SOAP (to communicate), Web services can be mixed and matched to create innovative applications, processes, and value chains.

This automatically means that only key areas of your application that need to be advertised and syndicated will need to become web services. The common types of web services that are beginning to emerge are – stock market quotes, currency conversions, locating used cars, dictionary lookups, temperature and weather conditions, Rent-a-car services, credit card validation, date and time services, flight Information etc.

Key Functions

Key functions are good candidates for web services. By discovering our key functions in the business processes we can discover possible candidates for web services. Asking a few general questions we can discover most of our key functions –

- What information do my customers / vendors / users of my system need?
- What are the relevant information that each of them commonly share?
- How do they require this information to be given?
- Who can see this information and what can they do with it? Can they change it or just access it?

Some Examples

An Inventory system may need to share its products stock levels with its customers and its raw materials stock levels with its vendors. This would enable transaction systems of the retail customers to place orders dynamically using a product purchase web services that results

in stocks being updated automatically and raw materials requirements going up proportionally.

How should my process be re-engineered?

So we now understand what a web service is and how our users can leverage it. But what do I change / re-engineer in my current process so that they can become web services? Well, it depends on who you are in this process – The Service Provider needs to enable his key services but the Users need to access web services. Let's take a closer look:

If I am a Service Provider…

The first job for a service provider would be to identify his key functions that need to be accessible to other users. Ensure that these functions have clear points of interface so that they can be interfaced directly to be given inputs by the user and to give outputs that are required by users. These functions then easily form web services. If a specific information group needs to be interfaced from the application then it becomes necessary to create the adequate interface from this collated data.

If I need other's Services…

As a user of other's services it becomes necessary to know what the kinds of services required are and what are available. Once that is clear, it becomes necessary to ensure that these services are reliable and fool proof. If you need to publish services that in turn depend on other web services then the quality of service needs to be ensured and this requires your services to be designed to reliability.