

Watch this: Web services

[Terrence A. Brooks](#)
The Information School
University of Washington
Seattle, WA 98195

What's a Web service?

The gist: Web services offer value-added information or information processing on the Web.

The implication: Share your expertise with the Web community.

The opportunity: You can publish both your information and information processing procedures for others to leverage in their programs.

An example: Agencies in your community might want to list your library hours on their Web sites. Their Web sites consume the Web service giving your library hours. If you change your hours, all the attendant Web sites automatically reflect the changes.

Suppose you're writing a computer program and you need a subroutine, a method or a procedure. You could write your own, but suppose the very procedure you need already exists on the Web compiled and ready to use. Being economical, you write your program so that it utilizes the distant procedure. You would be using a Web service.

Note the similarity of this model to active Web content based on legacy CGI (Common Gateway Interface) or ASP (Active Server Pages) technologies. Typically they accepted user input and responded with HTML (Hypertext Markup Language). Web services extend this model to the programming realm of application-to-application sharing. My application can use a procedure you wrote and make available on your server. Users of my program would be oblivious to the provenance of the operations occurring before them.

Historically, the desktop and the Web have been two different environments. Web services blur that distinction as desktop applications and Web applications can now share data and functionality. This is not an absolutely original concept: many Windows desktop applications right now will update themselves by automatically seeking new procedures and contents from the Web. You are required to OK the download, thus it doesn't occur invisibly in the background. Web services extend this sharing model two ways: (1) All of us can create Web services and bind them in our programs, and (2) The integration of program and Web service occurs silently in the background.

Key idea: Web services let applications share data and invoke capabilities of other applications.

Speculation: Is this the beginning of the [Semantic Web](#)? The vision of a Semantic Web suggests a distributed network of meaning. Web services would be a fundamental building block of such a vision because they

For further information:

[W3C Web Services](#) at the World Wide Web Consortium

[Understanding XML Web Services: The Web Services Idea](#) by Tim Ewald

[XML Web Services Developer Center Home](#) Microsoft site

[WS-Futures@Discuss.Develop.Com](#)
A Listserv of Web Service-related specifications

[Web Services and UDDI](#) An IBM perspective

[Web Services Architect](#) "An independent source of information about Web services"

[WebServices.org](#) Web services papers and more

Programming .NET Web Services Alex Ferrara & Matthew MacDonald.
Sebastopol, CA: O'Reilly, 2002
[ISBN: 0-596-00250-5]

provide the distant semantic functionality.

What's the advantage of Web services?

The most compelling reason to use Web services is leveraging a distant expertise in your own program. Why re-invent the wheel?

Web services are shared using the ubiquitous HTTP (Hypertext Transfer Protocol), thereby finessing the problem of heterogeneous operating systems and technology suites. Updates can be made at the server, without the need to distribute new DLLs (dynamic link libraries) to clients. Web services promise enormous efficiencies in the sharing of data and applications. Partisans of Web services, like Don Box of Microsoft, envision rebuilding the Web by replacing HTTP with SOAP ([Simple Object Application Protocol](#)) messages in place of the current packet-based system.

Ultimately, Web Services are just the next evolutionary step in software integration. If we do our jobs well and make Web Services an intrinsic part of the platform, then developers can go back to focusing primarily on building applications. I do believe that those applications will be considerably more interesting in a Web Service-based universe, primarily due to the decentralized nature of integration and development.
["Talking to...Don Box discusses Web services and his new role at Microsoft"](#) MSDN Magazine, vol. 17 (12) December 2002, p. 136.

Show me an example

UDDIs (Universal Description Discovery Integration) sites are the yellow pages of Web services. At this time the [UDDI at Microsoft](#) has about 850 listings. As of November 2002, my survey of Web services at the Microsoft UDDI reveals that many of them return simply the URL (Uniform Resource Locator) of a Web page, or address of a business. However, Web services directories are appearing such as [Remote Services](#) and [Web Services Mall](#)

Ironie comment: UDDI classifies Web services using classification schemes such as the North American Industry Classification System (NAICS - 1997), Universal Standard Products and Services Codes (UNSPSC-7.03) and the Standard Industry Classification (SIC-1987) and so on. Where is the evidence of the librarians, taxonomy experts, and indexers? Have they been ignored again by on-rushing technology?

Inhibitors of Web services

Can I really trust your Web service? Will your Web service persist in time?
In the future will you begin to charge for your Web service?

Key idea: Social aspects may inhibit the use of Web services. Examples of crucial limiting social aspects: Security threats and fear of too much centralization in the hands of one company.

Expect to see the efficiencies of Web services applied in communities of trust. An example would be a highly constrained corporate environment where work groups need to leverage the expertise within a large, distributed enterprise. For example, [Scott Tsao](#) of the Boeing corporation speaks of building a "community of practice." This is a company-wide community of

practice for XML in The Boeing Company that has a common culture, concept framework and trust based on knowing who designed and maintains distributed computer resources.

The bottom line: Watch Web services will flourish and manifest an aspect of the Semantic Web, but only in constrained communities of trust.

Date: November 2002

How to cite this paper:

Brooks, T.A. (2002) "Watch this: Web services" *Information Research*, 8(1) [Available at <http://InformationR.net/ir/8-1/TB0211.html>]
© the author, 2002 Updated: 14th November 2002

[Contents](#)

3 2 9 2
[Web Counter](#)

[Home](#)
