

# Go the

**A STEP BEYOND SERVERS  
AND BROWSERS, WEB-  
SERVICES TECHNOLOGY  
LINKS EMBEDDED  
DEVICES TO POWERFUL  
APPLICATIONS AND  
INFORMATION SOURCES  
ANYWHERE ON  
THE INTERNET.**

# distance

Illustration by Barroux

**A**S WE APPROACH the pervasive-computing vision, embedded appliances will take on a more comprehensive role, allowing users to

act on information from any place, at any time, and on any device. Web services are the key technologies behind this new role, which promises to give device users access to remote applications—along with time- and labor-saving services—independent of the hardware, operating system, or display. Although their approaches differ somewhat, IBM, Sun, Microsoft, and dozens of other large companies agree that Web services will revolutionize the use of the Internet and have developed extensive business strategies to capture a portion of the future market.

Whereas enterprise-software teams use them to streamline online business-to-business transactions, Web services will help embedded developers increase the functions of limited-resource systems. For example, a wireless handheld device could employ Web services to calculate the most efficient delivery route based on current traffic and road conditions. Likewise, a military computer can gather current information about the location of personnel and targets from other battlefield computers before firing a weapon system. A simple example of a Web service is a quotation service that accepts a stock symbol and returns the current price. All of these scenarios are possible today but require custom interfaces and software at both the server and the client devices. Web-services technology provides a standardized way to define the interface so that you can call new or legacy applications from any device, from any location, and at any time.

The recent proliferation of small, cheap processors along with low-cost wireless communications are an ideal setting for the growth of Web services in the embedded-system market. Web services fit perfectly with limited-resource embedded devices, because much of the computing horsepower resides elsewhere. The embedded device simply collects data from the user or the local sensors, communicates with one or more Web services, and then outputs or displays the results. Of course, Web services are not the answer in every instance, espe-

<i>At a glance</i> .....	<b>38</b>
<i>Mobile maps</i> .....	<b>40</b>
<i>For more information</i> .....	<b>42</b>





cially those in which transmission delays or occasional communications interruptions destroy their function. However, plenty of useful applications exist. You can also use Web services to enhance the primary function of a device by displaying useful but nonessential information.

Web services aim to break down the proprietary barriers between software. Almost every company has over the years developed a vast array of useful custom software to perform business and scientific tasks. A company seldom uses these programs elsewhere, because they are in obsolete or unpopular languages, the author has left the company, the platform is different, or any number of technical reasons. With Web-services technology, you can expose the functions and data of these applications as Web services, so that users within or outside the company may incorporate them into their applications or devices. Web-services technology allows devices to run and share applications and exchange information independently of the language, the platform, or the manufacturer.

**PLAIN TEXT, PLEASE**

To ensure widespread acceptance, the creators of Web services have decided to follow a strategy similar to that of the current Web-surfing technology. Web browsers have been highly successful because the communications are text-based and somewhat human-readable. Servers and browsers are machine-independent and have been developed in many languages and for most computing platforms. Web services also depend on the ability of the parties to communicate even if they are using different information systems.

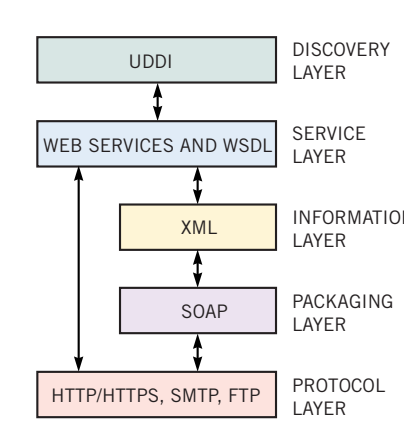
XML (eXtensible Markup Language) is a key technology in creating a platform-independent data-transfer procedure for Web services. Although XML encloses data within tags much like the HTML (HyperText Markup Language) tags that Web browsers use, significant differences exist. Although HTML tags specify how to present or display text, XML tags describe the contents of the enclosed text. Another major difference is that XML is extensible, meaning that you can define your own tags. With HTML, you are restricted to the tags that have been previously defined within the HTML specification. XML allows you to reference an existing file, called a

**AT A GLANCE**

- ▶ Web services allow networked embedded devices to execute remote applications as if they were part of the internal operating system.
- ▶ Open Internet standards—such as XML, SOAP, WSDL, and UDDI—ensure compatibility among services from sources worldwide.
- ▶ Partners and customers can share business information and order goods or services electronically over a Web-services link.
- ▶ Differences between Microsoft's .NET and Java 2 Enterprise Edition development tools may fragment Web-service interoperability.

schema, to define the data structure. You can also create a new one to define alternative tags and structures. The W3C (World Wide Web Consortium) defines a Web service as a software system identified by a URI (Uniform Resource Identifier) with public interfaces and bindings defined and described by XML. Other software systems may then interact with the Web service using XML-based messages that Internet protocols convey.

In addition to XML, several other Internet technologies are important to Web services, including the SOAP (Simple Object Access Protocol), the UDDI (Universal Description, Discovery, and Integration) format for application identification, and the WSDL (Web Services Definition Language). **Figure 1** shows how these technologies interact to form



**Figure 1** Web-service building blocks create a standard mechanism for identifying, describing, and calling remote applications.

a software stack for locating, describing, and executing a Web service. IBM, Microsoft, Oracle, and most other large software vendors support these technologies as the foundation of Web services. You can view and download the latest standards for these Web-services protocols and technologies from the W3C at [www.w3.org](http://www.w3.org).

SOAP is a simple request-and-response protocol that you use to exchange information in a distributed environment. It is an XML-based protocol for encapsulating messages that you send over the Internet. SOAP includes a message-description framework, data-type encoding rules, and a convention for remote procedure calls and responses. Although you can use SOAP to transfer messages over any protocol, the current specification emphasizes HTTP (Hypertext Transfer Protocol), the most common Internet transport protocol. SOAP messages use unique name spaces to identify and differentiate Web services on the Internet. Name spaces can be a domain name followed by a unique service name, such as <http://edn.com/subscribe>. Although many name spaces look like standard URLs, they need not point to actual pages on the Internet.

You describe a Web service's behavior using WSDL, an XML document that defines the message contents, Web location, and communications protocol. WSDL pulls together all the technical details you need for integrating the Web service into your application. It specifies argument types and return values that you can expect when invoking the service. A UDDI registry is a directory system that allows companies to publish a description of publicly available Web services. It is basically a phone book for Web services and includes business names, addresses, and contacts, along with addresses of Web services and interface information. Publishing information on a UDDI registry is optional, but it allows you to expose services to other business partners and customers.

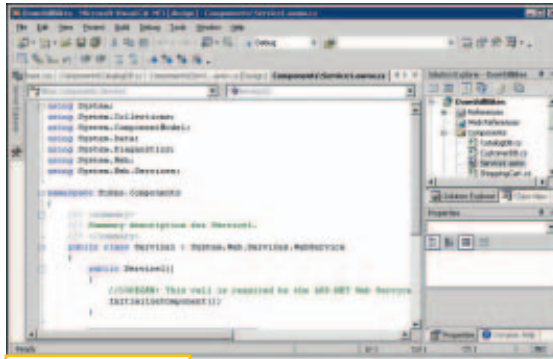
**SERVICE SECURITY**

One of the fundamental questions regarding Web services is transaction security. With thousands of hackers lurking in the background, how can businesses trust the Internet with their economic secrets? Initially, developers assumed that HTTP security techniques

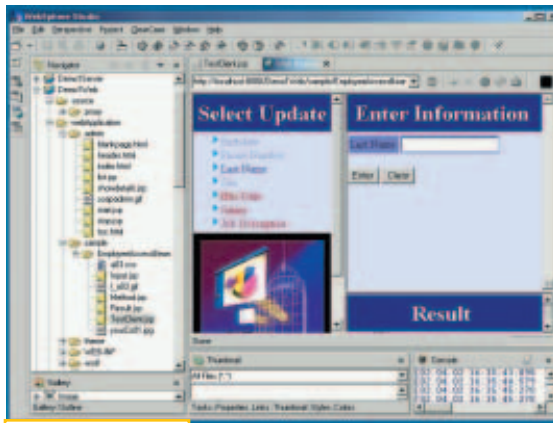


would be adequate, because thousands of server-to-client applications are already running with HTTP security. To add another level of security, IBM, Microsoft, and Verisign have proposed a Web-services-security specification that defines a standard set of SOAP extensions for additional security features, including XML signatures, encryption, and security tokens. You can find details of Web-services security at the Organization for the Advancement of Structured Information Standards Web site at [www.oasis-open.org](http://www.oasis-open.org).

To date, most Web services have been business-to-business applications on enterprise computer systems. Organizations are using Web services to link business applications and databases to other companies' systems via the Internet to improve customer-service operations and supply chains. Now, large software vendors and others are ready to extend Web services into the embedded world to be able to interact with data gathered from field operations (see sidebar "Mobile maps"). For example, as a company makes sales with small handheld point-of-sale devices, Web services could query pricing, direct inventory updates, download shipping schedules, and verify credit over a wireless Internet link. Embedded devices needn't only use or consume Web services; they can also provide them. For ex-



**Figure 2** Visual Studio from Microsoft integrates the tools for developing .NET-based Web services.



**Figure 3** IBM's WebSphere Studio offers Web-service development using the Java 2 Enterprise Edition environment.

ample, a service technician could remotely initiate self-test or diagnostics routines in an embedded system to identify parts for a service call, via Web services, from any device or desktop system.

Although most of the specifications

and protocols mentioned so far are open standards, plenty of proprietary products are available for Web-service development. Microsoft .NET, Sun One, and IBM WebSphere are all commercial development programs with unique products, architectures, platforms, and tools for building, testing, and deploying Web services. Sun, IBM, and several other software vendors have based their platforms on XML and Java; Microsoft's .NET framework emphasizes XML, Visual Basic, and a new object-oriented language, C# (pronounced "C sharp"), developed for Internet programming.

The language you use to develop a Web service is immaterial. However, incompatibilities among tool sets have created problems and debates. For example, Web-based applications built using .NET developer tools currently run only on Windows-based servers. Another Microsoft creation, Passport, is the user-authentication mechanism for .NET and has been controversial because a private company may become the identity gatekeeper of the Internet. Although Sun Microsystems and a consortium of 34 companies called the Liberty

Alliance has set out to create a competing way to verify identities, Microsoft has captured most desktop users. Microsoft's development tools are based on the popular Visual Studio environment modified for .NET and are available in Professional, Enterprise Architect, and Enterprise Developer versions (Figure 2). Visual Studio.NET prices start at \$1079.

#### WHAT ABOUT JAVA?

IBM includes Web-service development tools in its WebSphere Studio (Figure 3). You can create JAVA 2 Enterprise Edition (J2EE) and Web-services applications with integrated support for Java components, HTML, XML, and Web services, all in one development environment. You can build or enable applications with Web services, using UDDI, SOAP, and WSDL via a Web-services client wizard. WebSphere Studio, V4.0 Professional Edition cost \$496 and is available at the IBM Web site.

Sun Microsystems offers a starter kit,

## MOBILE MAPS

Web services are a new phenomenon in the embedded-system world, but working services to which you can subscribe are available. MapPoint .NET is an example of a programmable, XML (eXtensible Markup Language) Web service developed and hosted by Microsoft. This Web service is a platform for location-based services, such as delivery vehicles, and it allows developers to integrate location intelligence and mapping capabilities into mobile applications. You can call the MapPoint .NET platform from mobile applications, and it provides driving directions, shows maps, finds addresses and places, returns street addresses or geograph-

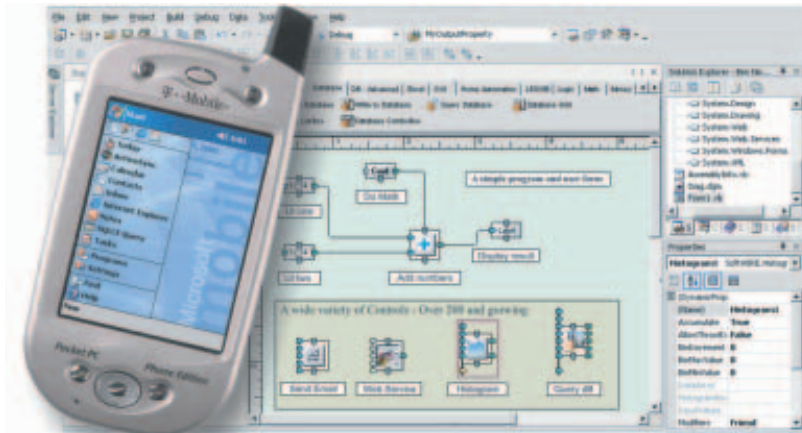
ic entities based on latitude and longitude coordinates, and performs proximity searches for points of interest.

MapPoint .NET is an XML Web service with a SOAP (Simple Object Access Protocol) API that finds locations, creates routes, renders maps, and offers a set of utility-function objects and methods that are common to all three of those tasks. To call the service, you need to reference the MapPoint .NET WSDL (Web Services Description Language). Subscribers to the Microsoft MapPoint .NET Basic Web Service pay a platform-access fee of \$15,000 plus a per-user license depending on volume.



including a preview of the soon-to-be-released Sun One Developer Platform, an integrated middleware stack for running and developing Web services and services on demand. The kit also includes the Sun One Studio 4 in community, mobile, and enterprise editions; the Sun One certificate, integration, and portal servers; the Sun One message queue; and Sun One active server pages. In addition, it includes the J2EE developer kit, the Java Web-services developer pack, and tutorials. The Sun One Starter Kit is available online for \$29.95.

In addition to the software giants, smaller vendors are readying products for the embedded-Web-services market. SoftWire for Pocket PC allows the creation of complete Pocket PC applications by graphically modeling the solution, as a flowchart, rather than coding it (Figure 4). The graphical interface lets nonprogrammers create powerful server-side Web



**Figure 4**

SoftWire for the Pocket PC allows nonprogrammers to create Web services using a graphical model.

services. SoftWire works with the Microsoft .NET Framework and sells for \$495 per seat for development. The applications you create and deploy require no runtime activation or licensing fee.

Off-the-shelf hardware- and software-development products can simplify the transition to a Web-services programming model. Intrinsic offers several developer kits supporting Microsoft's Windows CE.NET operating system. The CerfPod 250, for example, includes a 400-MHz Intel XScale processor, a touch-screen LCD, and network connectivity, and it comes installed with Windows CE.NET (Figure 5). The kit also comes with a Platform Builder board-support package, allowing you to configure the operating system for your exact needs. The CerfPod 250 development kit sells for \$1495, and you can order it directly from the vendor's online store.

Questra has developed a Web-services

product line integrating small, remote embedded devices as part of an enterprise-level distributed-computing architecture. Questra's products feature a smart agent installed on an embedded device at an end-user site. A configuration engine lets you send Web-services protocols to and from the device. Basing its software on J2EE, Questra has developed a set of Web-services test suites to measure new hardware devices, interfaces, and performance differences. The Questra Smart Service product is a small-footprint application that you download onto the equipment, to ease the gathering and transmission of real-time data, and an Internet-monitoring device that collects the data and transmits it to an enterprise location. Questra has also experimented with very small, embedded devices, trimming the Web-services memory overhead to 20 kbytes for minimal XML and SOAP support.

Web services promise to change the way you design embedded devices, allowing powerful functions in a limited-resource system. With a simple but standardized software mechanism for transferring and receiving parameters over a network, your device can link to applications all over the world. Technologies such as XML, SOAP, WSDL, and UDDI, along with development tools from Microsoft, Sun, IBM, and others, give you everything you need to add Web services to your next embedded design. □

You can reach Technical Editor Warren Webb at 1-858-513-3713, fax 1-858-486-3646, e-mail [webb@edn.com](mailto:webb@edn.com).




**Figure 5**

The CerfPod 250 developer's kit from Intrinsic supports Microsoft's Windows CE.NET operating system.

### FOR MORE INFORMATION...

For more information on products such as those discussed in this article, go to [www.edn.com/info](http://www.edn.com/info) and enter the reader service number. When you contact any of the following manufacturers directly, please let them know you read about their products in *EDN*.

**IBM**  
1-800-426-4968  
[www.ibm.com](http://www.ibm.com)  
Enter No. 301

**Intrinsic Software**  
1-604-801-6461  
[www.intrinsic.com](http://www.intrinsic.com)  
Enter No. 302

**Microsoft**  
1-425-882-8080  
[www.microsoft.com](http://www.microsoft.com)  
Enter No. 303

**Oracle Corp**  
1-650-506-7000  
[www.oracle.com](http://www.oracle.com)  
Enter No. 304

**OTHER ORGANIZATIONS MENTIONED**  
**Organization for the Advancement of Structured Information Standards**  
[www.oasis-open.org](http://www.oasis-open.org)

**Questra Corp**  
1-585-381-0260  
[www.questra.com](http://www.questra.com)

**SoftWire Technology**  
1-508-946-8900  
[www.softwire.com](http://www.softwire.com)

**Sun Microsystems**  
1-650-960-1300  
[www.sun.com](http://www.sun.com)

**Verisign Inc**  
1-650-426-5115  
[www.verisign.com](http://www.verisign.com)

**World Wide Web Consortium**  
[www.w3c.org](http://www.w3c.org)

#### SUPER INFO NUMBER

For more information on the products available from all of the vendors listed in this box, enter no. 305 at [www.edn.com/info](http://www.edn.com/info).