

New Standard Closes the Loop for Web Services

By Ed Anuff

New standards are emerging to let companies build applications that work together using eXtensible Markup Language (XML) messaging. The standards include:

- Simple Object Access Protocol (SOAP)
- Web Services Description Language (WSDL)
- Universal Description, Discovery, and Integration (UDDI)
- Web Services User Interface (WSUI).

These standards provide a good foundation for machine-to-machine communication and will help XML Web services usher in the next-generation Internet. WSUI is especially valuable because it addresses the lack of a standard user-interface layer for Web services. The proposed WSUI standard lets Web services with SOAP-Remote Procedure Call (RPC), XML, or WSDL SOAP interfaces be described as complex, multi-page, end-user applications that can be easily embedded in Websites.

WSUI leverages the benefits of an XML-based interface to an application — benefits that make Web services an ideal machine-to-machine transport layer. Among the benefits are separating implementation from interface and enabling diverse implementation platforms to interoperate. By standardizing the display layer of Web services, vendors can more easily embed their applications in Web application delivery platforms without expensive, vendor-specific implementations.

WSUI leverages existing standards and adds a new technology — the WSUI

schema — for vendors to support in order to easily share applications. A WSUI component represents a single application — such as a stock quote, shipping tracker, or calendar application — that can be exposed as a series of multi-page views and contain embedded workflow. By embedding a WSUI implementation, vendors of Website publishing platforms can easily integrate WSUI services without lengthy integration efforts of (or even knowledge of) third-party services and applications. Customers benefit from:

- The ability to expose custom-built applications as WSUI services for embedding in internal Websites and partner Websites
- Not being limited to any one application development platform (since WSUI separates application implementation platforms from application delivery platforms).

Solving the Business Problem

The objective of WSUI is to create an easily implemented standard to enable Web services to be integrated at runtime into a user interface for displaying a Web application to an end user. WSUI proponents are highly experienced in integrating multiple Web applications as embedded components of other Web service

publishing platforms. Some vendors have created proprietary JSP, ASP, or XML SDKs to expose their application functionality for integration into third-party Websites. However, the lack of a standard mechanism for exposing Web service applications has meant that most work done by vendors and customers cannot be leveraged more than once. By building a WSUI interface to their applications, application and service vendors can expose the applications to any platform that supports WSUI. Vendors achieve lower integration costs while customers benefit from reduced implementation costs and greater interoperability.

The WSUI Philosophy

Creating a standard user-interface layer for Web services is challenging. It requires making the standard general enough that it can be:

- Implemented relatively easily on disparate server platforms such as Java, .Net, and Perl
- Flexible enough to enable real business applications to be exposed.

WSUI leverages the basic XML technologies supported in open-source toolkits such as the Apache XML project (<http://xml.apache.org>) as well as vendor toolkits that implement core XML specifications. WSUI is specifically engineered to expose components, which expose any number of end-user views in a browser. Usually, the purpose is supporting form and menu-driven user interaction with back-end services. A WSUI component can call XML, SOAP RPC, and WSDL SOAP services. More service

WSUI leverages existing standards and adds a new technology.

types can be easily added in the future. WSUI's creators acknowledge that many Web services will never be interacted with via a user interface. Nevertheless, many services that end users and customers want to aggregate are user-facing. That's where WSUI is beneficial because it offers a standard mechanism for integrating user-facing services via standards-based XML messaging.

WSUI Solution Architecture

WSUI makes several key assumptions about the interested parties. A service provider exposes any number of XML or SOAP services. A service consumer is a Web publishing platform or portal, which is running a WSUI container implementation and constructs Web pages based on user requests. Users interact with Web pages to interact with a WSUI service (see Figure 1).

Architecturally, WSUI assumes that the services exposed via XML or SOAP exist as-is. For a Web publishing platform to integrate the service as an application that can be accessed by end-users requires only a WSUI descriptor file plus a series of XML Stylesheet Language Transformation (XSLT) templates. Developers can create the actual WSUI file and XSLT templates needed to integrate existing services. An application or service vendor can package the WSUI/XSLT files with a standard interface to facilitate other Websites embedding their applications.

User Case Scenarios

Scenario #1: Shipping Tracker

Shipping trackers are a common example of a Web service where the application logic resides on a remote server. A shipping tracker is typically a simple two-stage application composed of a front-page form for entering a tracking number and receiving a result view. Websites that support personalization may support a slightly more complex interaction model whereby a personalization view lets a user enter a tracking number(s) and the front-page view automatically displays the shipping status. This is an example of a Web service application that, using WSUI, can be delivered as a hosted service over the Internet for embedding in other Websites.

Scenario #2: ERP Application

An Enterprise Resource Planning

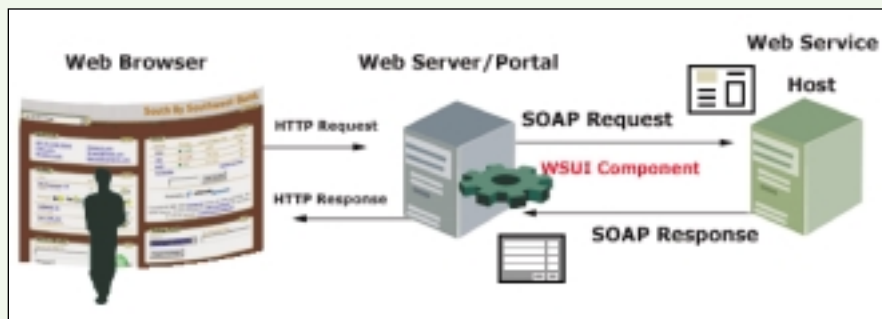


Figure 1

(ERP) application provider might wish to expose a series of modular page components to reveal their functionality for embedding in other Websites. For instance, an HR application might expose a series of views such as "clock in/clock out," "anniversary list," "employee overview," and "record working time." Previously, exposing such functionality has been done via client-side Javascript or ActiveX plug-ins that don't offer much display flexibility to delivery platforms and cannot be customized. By using WSUI, these components can easily be integrated into any WSUI-compliant Web platform. This is an example of a Web service application that can be delivered as a packaged interface to a new or existing software product that needs to be delivered as a packaged component, embedded in other Websites via WSUI.

Scenario #3: Custom Enterprise Application

A custom enterprise application often needs to be integrated into portals and Websites in other offices or operated by other units of the organization. For instance, an expense tracking application written in Cold Fusion might need to be integrated into a Website running on a Java-based platform. By creating a SOAP interface to the application and then adding a WSUI layer, all other internal Web application platforms can dynamically "subscribe" to this service. This dramatically reduces the costs of integration and lets each department run its Web operations on its preferred platform. This is an example of using WSUI to expose services of a custom corporate application as a Web service application to Websites within a corporate network.

Conclusion


WSUI offers several benefits to vendors, customers and developers:

- It's a standard mechanism for packag-

ing Web services as user-facing applications.

- It leverages existing business and technical standards. It's easily implemented on any robust server-side platform that supports XML.
- It's an open specification and implementation the Web services community can review and use.

WSUI eases the packaging and embedding of multi-stage Web applications in other Websites using a standards-based approach. It leverages existing Web services standards. By adding WSUI as a component of the Web services stack provides a more effective approach to Web application integration. That approach is essentially to "develop once and share many times." In this way, WSUI:

- Eases integration
- Avoids dependency on any one vendor's solution
- Provides a flexible, clean approach to creating Web services and applications
- Is a comprehensive framework for creating services and integrating them into a user-facing application. 

About the Author



Ed Amuff is chairman, co-founder, and chief strategy officer of Epicentric, Inc., a leading provider of business portal solutions and the primary organizer behind the

WSUI initiative. Before co-founding Epicentric, he launched and managed the popular HotBot search engine while director of product management at Wired Digital. He is also the author of the Java Sourcebook (*J. Wiley and Sons*), one of the first books on the Java programming language. e-Mail: ed@epicentric.com; Website: www.epicentric.com.