

Chapter 7.10

User Facing Web Services in Portals

Jana Polgar
NextDigital, Australia

ABSTRACT

In SOA framework, Portal applications aggregate and render information from multiple sources in easily consumable format to the end users. Web services seem to dominate the integration efforts in SOA. Traditional data-oriented web services require portlet applications to provide specific presentation logic and the communication interface for each web service. This approach is not well suited to dynamic SOA based integration of business processes and content. WSRP 2.0 aim at solving the problem and providing the framework for easy aggregation of presentation services. Is not practical to publish portlets locally if the organisation wishes to publish their portlets as web services to allow their business partners using these services in their portals. UDDI extension for WSRP enables the discovery and access to user facing web services while eliminating the need to design local user facing portlets. Most importantly, the remote portlets can be updated by the web service providers from their own servers.

VISION FOR USER-FACING PORTLETS

Web services introduced the means for integrating and sharing business processes via the Internet. WSRP (WSRP specification version 1 (2003)) goal is to extend the integration further by providing framework for sharing web service presentation components. WSRP specification formulated a standard protocol which enables all content and application providers to create web services, generate their presentation faces as HTML fragments and offer them to the consumers to be plugged into their local portals.

Portals and portlets (JSR 168 (2005)) provide specific presentation logic to aggregate data from multiple sources which could be legacy systems, Enterprise Information Systems (EIS), local or remote web services, or EIS with exposed web service interfaces. The first draft of JSR 286 (JSR 286 (2008) brings new features to the Java portlets capabilities introduced by WSRP 2.0 (WSRP Specification version 2.0 (2008)). JSR 286 new features include:

- Interportlet communication: coordination between portlets and allow building composite applications based on portlet components;
- Shared render parameters enable to specify which render parameters they can share with other portlets;
- Resource serving feature enables portlets to serve resources within the portlet context;
- Frameworks for better support for JSF and Struts
- Alignment with WSRP 2.0
- Better user experience using AJAX patterns
- Portlet filters to selectively define the portlets which can transform the content of portlet requests and responses on the fly.

The WSRP specification is intended for presentation-oriented web services, user-facing web services that can be easily integrated with portals. They let businesses provide content or applications without requiring any manual content or application-specific adaptation by portal presentation logic. It is envisaged that in the near future portals will easily aggregate WSRP services without any programming effort. The only effort required is the actual deployment of remote portlets in the local portal server (Hepper, S and Hesmer, S. (2003)). We are not taking into account the effort needed for the “implementation”, that is the design of the portal page which is needed in any case.

The WSRP specification (WSRP specification version 1 (2003) and WSRP 2.0 are the effort of the working group at OASIS (<http://www.oasis-open.org/committees/wsrp>). It aims to provide a set of options for aggregating user-facing web services (remote portlets) from multiple remote web services within one portal application. WSRP standard has been conceived for implementing simple services. The developer of the portlet provides the markup fragments to display web service data. The current version allows for more

complex services that require consumer registration, support complex user interaction, and operate on transient and persistent state maintained by the service provider. Before looking at the functionality of WSRP, note that what WSRP refers to as a portlet is the combination of a portlet implementation and any configuration data that supports the implementation. WSRP 2.0 (WSRP Specification version 2.0 (2008) is closely aligned with the JSR286 thus providing the framework for publishing JSR286 portlets as web services.

WSRP AND WSRP RELATED STANDARDS

WSRP defines the notion of valid fragments of markup based on the existing markup languages such as HTML, (X)HTML, VoiceXML, cHTML, etc (Figure 1). For markup languages that support CSS (Cascading Style Sheet) style definitions, WSRP also defines a set of standard CSS class names to allow portlets to generate markup using styles that are provided by WSRP compliant portals such that the markup assumes the look and feel of the consuming portal.

WSRP is fully integrated with the context of the web services standards stack. It uses WSDL additional elements to formally describe the WSRP service interfaces and requires that at least SOAP binding be available for invocations of WSRP services. WSRP also defines the roles of web service *producers* and *consumers*. Both *producers* and *consumers* use a standard protocol to provide and consume web services for user facing portlets. The WSRP specification requires that every *producer* implement two required interfaces, and allows optional implementation of two others:

1. **Service Description Interface (required):** This interface allows a WSRP *producer* to advertise services and its capabilities to

17 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/user-facing-web-services-portals/37727

Related Content

Web Data Warehousing Convergence: From Schematic to Systematic

D. Xuan Le, J. Wenny Rahayu and David Taniar (2006). *International Journal of Information Technology and Web Engineering* (pp. 68-80).

www.irma-international.org/article/web-data-warehousing-convergence/2619

Introduction to Performance Monitoring and Tuning: Java and .NET

B. M. Subraya (2006). *Integrated Approach to Web Performance Testing: A Practitioner's Guide* (pp. 234-269).

www.irma-international.org/chapter/introduction-performance-monitoring-tuning/23981

A Proposed Template for the Evaluation of Web Design Strategies

Dimitrios Xanthidis, David Nicholas and Paris Argyrides (2010). *Web Technologies: Concepts, Methodologies, Tools, and Applications* (pp. 119-144).

www.irma-international.org/chapter/proposed-template-evaluation-web-design/37628

Sizing Web Applications

Emilia Mendes (2008). *Cost Estimation Techniques for Web Projects* (pp. 78-119).

www.irma-international.org/chapter/sizing-web-applications/7163

An Effective Recommender System Based on Clustering Technique for TED Talks

Faiz Maazouzi, Hamed Zarzour and Yaser Jararweh (2020). *International Journal of Information Technology and Web Engineering* (pp. 35-51).

www.irma-international.org/article/an-effective-recommender-system-based-on-clustering-technique-for-ted-talks/241775