WSRP PFB Abstract Model

Version

1.0

OASIS WSRP TC

Version Created

18th August 2004

Document Identifier

wsrp-pfb-abstract-model-1_0.doc

Editor

Andre Kramer, Citrix Systems Inc. (Andre.Kramer@eu.citrix.com)

Previous Editors

Alan Kropp, Vignette Corporation (akropp@vignette.com)

Preface

This document provides guidelines for publishing, finding and binding WSRP Producers and Portlets as *Services*, in terms of an abstract publishing model. The Publish, Find, Bind (PFB) Abstract Model presented is fully compatible with the OASIS WSRP 1.0 open standard and provides a framework for publishing WSRP Producers and Portlets in specific *Service registries* and *indexes*. While this model itself is not normative, specific concrete schemes for advertising and discovering WSRP Producer and Portlet services (for example, interoperable use of UDDI or ebXML Registry) are expected to follow the PFB Abstract Model and to adopt the guidance presented here. This document may be read as support material for the Registry specific technical notes produced by the WSRP TC.

Terminology

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in RFC2119.

PFB Overview

Publishing, Finding and Binding (PFB) supports the lifecycle requirements of WSRP Producers and Portlets, when advertised and discovered as *Services*. The Web Service Architecture developed by the W3C Web Service Architecture Working Group is a good source of background information on Web Service Discovery. WSRP 1.0 already provides support for this lifecycle by:

- Providing a full WSDL description of WSRP Producer services: WSRP Producers are encouraged to provide all available bindings as WSDL services. The default SOAP binding for the WSRP required ports (`WSRP_v1_ServiceDescription_Binding_SOAP` and `WSRP_v1_Markup_Binding_SOAP`) MUST be present.

- Exposing the ServiceDescription (`WSRP_v1_ServiceDescription_PortType`) port type, which MUST publish all metadata for a WSRP Producer and the Portlets that it offers (see the `ServiceDescription` and `PortletDescription` types from the WSRP 1.0 specification's XML Schema). In addition, the optional `getPortletDescription` method allows metadata for a specific portlet to be obtained.

PFB builds on this support and provides a light-weight framework for publishing to various Service Registries and Meta-data discovery services and indexes. Note that, our PFB Abstract Model does not prevent publishing the above information in other ways or forms.

No direct use of the PFB Abstract Model is made by end users who deploy and/or import WSRP services in Portals. Typically, end-users (for example, Portal Administrators) utilize PFB tools that conform to a PFB Technical Note for a specific service registry technology, which in turn follows the guidelines established by our abstract publishing model. Technical notes, including UDDI and ebXML Registry, are available from the WSRP TC Web pages.

PFB Use Cases

Base Use Cases

Two base use cases, as analyzed for PFB, through separate application domain/Business Scenario specific use case write-ups, support publishing WSRP Producers (WSRP Business Scenario Search UDDI Registry for Producers) and Portlets (WSRP Business Scenario Search UDDI Registry for Portlets) as services. In these use cases, basic standard support for publishing and finding is envisaged for users of a registry specific technology. Published services are augmented with extra information (metadata such as categorizations or policy) so that they can be advertised and discovered in domain specific ways. This document describes the basic standard model (in the Information Model sections) underlying these use cases and identifies ways that it can be extended (Model Augmentation section).

Query Use Cases

Application domain independent support for categorizing Producers and Portlets, optionally by specific WSRP version(s), and enumerating Portlets exposed by a Producer are addressed by our Query Use Cases. These should be supported by a registry technology as prioritized in Appendix 1.

Analyzing PFB Requirements

Our abstract model seeks to publish the least amount of WSRP related information required for service discovery into Service Registries. The information published by our model can and should be *augmented* with extra metadata as desired for a particular PFB scenario (as in the base use cases developed for PFB, for example). However, such extra information is not required for interoperable service discovery.

Our abstract model does specify the informational items that must be published, and that are pre-requisites when discovering (finding, selecting and binding) WSRP Producers and Portlets as Services. These items differ somewhat for the Producer and Portlet base use cases (see the Information Model sections).

A non-functional requirement for the abstract model is simplicity of publishing, as well as discovery[1]. In particular, it is desirable that WSRP Producers can be published as services directly by Administrators by hand, for example, using a service registry GUI or Web UI. This requirement is less important for Portlets, as the common use case will be to publish them in batches, based on the WSRP Producer's service description metadata.

Our basic use case is publishing WSRP Producers as services and the PFB Abstract Model can be used just for this purpose. This implies that publishing Portlets is optional, even if the Producer of the Portlets is published.

It must be possible to discover both the default SOAP bindings, for HTTP and/or HTTPS (HTTP over SSL), and additional protocol bindings, as well as any for additional interfaces. We require the Web service description to be provided by a service WSDL document and for this description to be *authoritative*. Binding information can also augment a producer service as described in the next section.

Diverse kinds of information may augment our PFB Abstract Model. It is through such augmentation of our basic service information that new or advanced use cases for PFB can be supported. Therefore, we actively encourage the use of additional metadata, but seek to avoid duplication of metadata already available via standardized WSRP and WSDL interfaces.

Categorizations supported by our abstract model should use standard identifiers recommended by a PFB technical note for a specific registry technology when possible.

The link between Portlets and their WSRP Producer should be maintained when publishing Portlets as services.

[1] WSRP Consumers already deal with the issues of selecting service bindings from WSRP 1.0 service WSDL documents.

Model Augmentation

Before describing the exact service publishing information that comprises our PFB Abstract Model, we outline some common types of additional metadata that we envision may augment our core PFB informational model.

Protocol specific binding information may be provided directly with a published Producer or Portlet service advertisement. We view the bindings described in a WSRP Producer's service WSDL as authoritative (see wsrp_service.wsdl as a non-normative example of a *service wsdl*). However, WSRP services are free to publish additional binding information for the WSRP standard port types, as well as other port types implemented by the Producer or Portlet service[2]. These can be used directly, without consulting the authoritative WSDL document at the WSRP Consumer's discretion.

Producer and Portlet metadata, as obtained via the `ServiceDescription` port type. This duplicates the information already available via the WSRP interfaces themselves, at the risk of update incompatibilities. Such information should be used only to support more direct query over published services. An example of such a metadata based query, could be finding all portlets that have the same "`groupID`" identifier in their service description, or filtering out any Portlet that "`usesMethodGet`" from a portlet selection user interface dialog.

Additional categorization information. Like any Web Service, we envision that WSRP services, both Producers and Portlets, will be *categorized* into vertical application domains (e.g. Medical dosage Portlets under Healthcare as in our Portlet use case) or by additional horizontal attributes (such as service's physical location or payment model). The abstract model's use of categorization only serves to support the basic set of queries identified by our query use cases (see the Appendix 1).

Additional **Service metadata**, such as Policy descriptions, that is not available through the WSRP service description, except possibly as extensions (see section 5.1.1. in WSRP 1.0). This may be in both machine and human readable forms.

[2] While a WSRP Producer or Portlet service may logically have other Web Service interfaces (port types) associated to it, the association and how the relationship is represented is outside the domain of both our PFB Abstract Model and WSRP.

The Information Model

The PFB Abstract Model[3], as developed based on use case analysis, directly supports the publishing of both WSRP Producer and Portlets as services. Items in bold typeface in this section are informational items in our information model (see also Appendix 2 for a summary).

We encourage all Portlets offered by a Producer to be published individually as services, but note that publishing Portlets is optional[4] in the PFB AM. Furthermore, a producer should not publish Portlet services to a registry that are not also described via "offered Portlet" WSRP metadata (`offeredPortlets[]` in the full WSRP service description `ServiceDescription`) or are optionally broken out as portlet descriptions (`PortletDescription`) returned by the `getPortletDescription()` operation in the optional Portlet Management port type. If available, `getPortletDescription()` can be used to complement the metadata returned in the `ServiceDescription`. This avoids WSRP Consumers having to deal with additional ways of obtaining metadata, and fits our philosophy of having the service data be authoritative.

Both Producers and Portlets are published under a **service name** in some business or organization scope. The business or organizational scope is that of the entity logically publishing and owning the service. The human readable service name should be fully internationalized. An optional short **service description** should also be provided and internationalized. The remainder of the abstract publishing information is dependent on the type of WSRP service (**producer service** or **portlet service**).

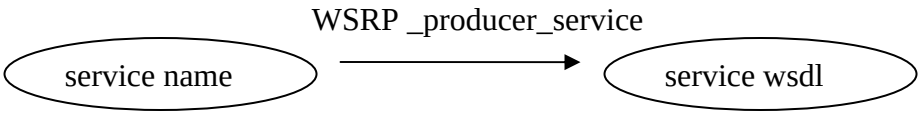
The representation of the owning business or organizational scope is highly dependent on the service registry and not further specified by the PFB AM. A portlet service and its producer may be published in different business or organizational scopes. This scenario is common in the Application Service Provider (ASP) model.

[3] Henceforth abbreviated to PFB AM.

[4] As use of PFB is itself completely optional, optional here means using the PFB Abstract Model but not publishing any Portlet services. Producers are not optional in the Abstract Model, in that all Portlets require a back link to their Producer.

Producer Service specific information

In the PFB AM, a WSRP producer is published as a service by associating a producer service WSDL document (**service wsdl**) with a service name. On its own, this would simply serve to publish some abstract WSDL based Web Service. The exact relationships to WSDL based Web Service description can be explored further in a concrete registry technology[5], but our abstract model only requires one specific relationship:



The **WSRP_producer_service** predicate (in RDF terms) relates a service name (the predicate's subject) to its WSRP service WSDL document (the object). It is to be represented by a categorization or type classification of the concrete service registry the producer service is published in.

A concrete service registry technology can decide to capture the relationship either using:

- A generic predicate valid for all versions of WSRP. In this case the WSDL object needs to be inspected for version information as well as service location information in the usual way.

- Multiple version specific predicates, one for each WSRP version supported by a published producer.

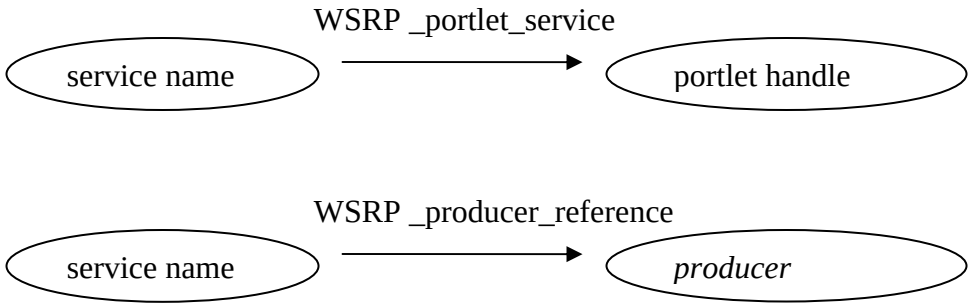
If the latter approach is taken then publishers should publish a producer service under all the WSRP versions that it supports and consumers should discover services by looking for specific versions. This avoids any dependency between version relationships at the PFB AM level.

For the former approach, remember that WSRP port types communicate version information using a `"WSRP_V" + Version + "_" + PortTypeName + "_PortType"` naming convention. This convention is understood by WSRP Consumers and directs the discovery of compatible bindings in a WSDL document.

[5] We encourage the use of the WSRP binding WSDL document in Web Service categorization (i.e. referencing wsrp_v1_bindings.wsdl).

Portlet Service specific information

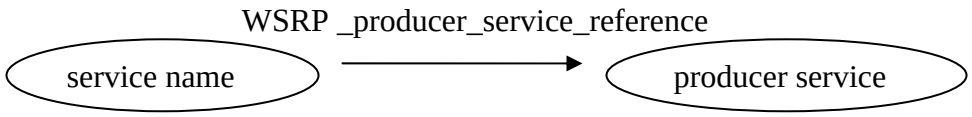
A **portlet service** is represented by the following relationships:



The **WSRP_portlet_service** predicate serves to identify a published service as a WSRP Portlet service. Unlike the corresponding WSRP_producer_server reference, which can be version specific or version neutral, this predicate should be version neutral. It is used to associate a portlet service with the **portlet handle** used to identify one of the portlets in a WSRP Producer (holds the value of the `portletHandle` element in a `PortletDescription`).

The above is not sufficient to allow a WSRP consumer to bind to the portlet service as the WSRP Producer hosting the portlet, corresponding to the portlet handle, needs to be identified. This relationship is modeled by our **WSRP_producer_reference** predicate. As can be seen from the diagram, it is used to associate a *producer* with the portlet service.

It is strongly recommended that each portlet service is associated with its producer. The exact format of this abstract reference to a producer is not specified by our PFB AM. Different types of *producer reference* are allowed, so that both *inter-registry* and *intra-registry* publishing relationships can be modeled. However, a certain class of producer reference is explicitly captured by our abstract model:



A **WSRP_producer_service_reference** relates a portlet service to a producer service that is published in the same concrete service registry. All concrete registry technology usage should support such a relationship for modeling the common case where a portlet service is hosted by a producer service in the same registry.

Security

WSRP Services should be advertised and discovered in a secure manner. The right to read and query a registry may be required to be restricted, as well as the administrative publishing (write or create/update/delete) rights. A registry may provide a different view, of the services published, to a specific user, so service discovery should be performed in the context of a specific user identity or user role, possibly that of a WSRP Consumer administrator. The service description itself must be protected, especially if it contains binding information for secure protocols (i.e. its integrity needs to be verifiable).

Appendix 1 – Query Use Cases

Here we list the query use cases that are supported by our PFB Abstract Model, in addition to "find service by name". Note that, 1 and 3 are equivalent to 2 and 5 while we only have one version of WSRP defined (i.e. 1.0) and the service registry technical notes should take account of this, while being future proof with respect to new versions (such as the expected WSRP 2.0).

1. Ability to locate all WSRP Producers, regardless of WSRP version(s) supported.
2. Ability to locate all WSRP Producers supporting a particular WSRP version.
3. Ability to locate all WSRP Portlets, regardless of WSRP version(s) supported.
4. Ability to locate all WSRP Portlets hosted at a particular WSRP Producer.
5. Ability to locate all WSRP Portlets supporting a particular WSRP version.

The ability to locate all WSRP Portlets offered at a particular Producer (4) is already covered by the WSRP offered Portlets (`offeredPortlets`) enumeration in output messages of the `getServiceDescription()` operation from the Service description port type, but may be directly supported by a concrete service registry for efficient query. (5) may also be supported only indirectly.

Searching for producer services is based on the WSRP_producer_service predicate. Finding portlet_services should leverage the WSRP_portlet_service predicate.

Appendix 2 – Abstract Model Informational Items

service name – human readable text, internationalized. Unique in a business or organizational scope.
service description [optional] – a short textual description of the service (multiple locale specific versions may be provided).
producer service – a WSRP Producer published as a service.
WSRP_producer_service – classifies a service to be a WSRP Producer.
service wsdl – a WSRP service WSDL document. Containing fully qualified service elements for WSRP required and optional port types.
portlet service – a WSRP Portlet published as a service.
WSRP_portlet_service – classifies a service to be a WSRP Portlet.
WSRP_producer_reference – associates a portlet service with its *producer*.
WSRP_producer_service_reference – associates a portlet service with its producer service.
portlet handle – identifies a WSRP portlet with its producer.
business scope – not fully specified, an honourable member of our PFB AM.

Appendix 3 – Notices

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.