WSRP – UDDI Technical Note

Version

draft-05

WSRP Publish Find Bind SC

Created

12/09/2003

Document Identifier

wsrp-pfb-uddi-tn-draft-05.doc

Editors

Richard Jacob, IBM (richard.jacob@de.ibm.com)
Andre Kramer, Citrix Systems Inc. (Andre.Kramer@eu.citrix.com)

Table of Contents

1 Preface

This document proposes a standard methodology for publishing and finding WSRP Producer and Portlet services in a UDDI registry. It is not the purpose of this document to educate on WSRP [1], WSDL [4] or UDDI [5,6]. The reader of this document should have a good understanding of the WSRP specification, the WSDL specification and the UDDI specifications and data structures.

2 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]***.

Compliance: Mandatory – relevant to legal rules, regulations or laws. Compliancy is the act of complying with a specification and/or standard. Example: ISO 9001. IEEE defines as complying with laws and regulations.

Conformance: Not mandatory – ISO/IEC Guide 2 defines conformance or conformity as fulfillment of a product, process or service of specified requirements.

[ak: not sure we need compliance and conformance definitions.]

3 Conventions

Throughout the document XML code fragments are employed to define the data structures used. The following text formatting conventions are used to aide readability:

- Key value Placeholders
  Data structures may contain values which reference UDDI key values such as tModel keys, service keys and business keys. These keys uniquely identify the data structures within the UDDI repository. For convenience and better readability, these key values are replaced by meaningful textual variables to represent such keys. Another reason for using the placeholders is that some key values are not know yet, as the tModel keys are generated by the UDDI directory once a tModel is published and actual values may vary between registries.
  For example, the following placeholder refers to a tModel key identifying the WSRP_PRODUCER tModel:

```
<tModel tModelKey="${TMODELKEY_WSRP_PRODUCER}" >
```

- Constants

Constant values are printed in the `Courier New` font always, regardless of whether they are defined by the UDDI technical note or this document. In addition, constant values defined by this document are printed using **`bold face`**. The following example shows the tModel name defined by this proposal for the **WSRP_PRODUCER** tModel:.

[AK: WSRP_PRODUCER should be bold and the access point URL should be italic but I was not able to make Word do this.]

```
<tModel tModelKey="${TMODELKEY_WSRP_PRODUCER}" >
  <name>
        WSRP_PRODUCER
  </name>
```

- Example Values
  These values are represented in *italic* font. In the following, an access point example value is shown:

```
        <accessPoint useType="http">
            http://vendor.com/producer/ServiceDescription
        </accessPoint>
```

4 General Concepts

4.1 WSRP Actors

The WSRP specification describes three major actors relevant to this technical note. WSRP “Producers” are presentation-oriented Web Services that host “Portlets” which are able to render markup fragments and process user interaction requests. “Consumers” use these Web Services to aggregate and present the generated markup to End-Users and manage the user’s interaction with the Portlet markup.

4.1.1 Producers

Producers are modeled as Web Service containers hosting and managing Portlets. Producers provide Web Service interfaces (PortTypes) that allow Consumers to interact with the Producer and especially with Portlets hosted by that particular Producer (see [add ref here]).

4.1.2 Portlets

Portlets are hosted by Producer Web Services and generate markup and process user interactions with that markup. In general, a Portlet includes both code and a particular configuration of any settings or properties the Portlet exposes. Portlets are addressed by a “Portlet Handle” scoped by the Producer environment.

4.2 WSRP PortTypes and Bindings

The WSRP interfaces are factored into multiple PortTypes. Each PortType provides a certain functionality subset of the full WSRP protocol. Some PortTypes are optional. Each PortType is defined in the WSRP Interfaces WSDL file (see [ref to Interfaces WSDL] for v1 PortType definitions).

The following PortTypes are defined by the WSRP v1 specification:

1. **Service Description PortType**
   This PortType provides a means for the Consumer to discover capabilities of the Producer and its Portlets. This metadata MAY be context sensitive, for example, applying to a particular consumer registration. The Service Description PortType is mandatory.
2. **Markup PortType**
   WSRP compliant services MUST implement and expose this PortType which deals with the generation of markup and the user interaction with this markup.
3. **Registration PortType**
   Producer MAY choose to support in-band registration of Consumers and then must implement this PortType.
4. **Entity Management PortType**
   In addition to “Producer offered Portlets” available through the service description, the Producer MAY expose this interface to allow Consumer to clone and customize Portlets. Such Portlets are referred to as “Consumer offered Portlets”.

For each of the above PortTypes, WSRP Producers define one or more Bindings. A Producer must define a default SOAP Binding using HTTP or HTTPS as the transport mechanism.

4.3 WSRP WSDLs

The WSRP Web Service description is factored into three parts. Firstly, the Types XSD document defines the types used by the WSRP protocol. Secondly, the Interface WSDL document defines the PortTypes and their operations. It imports the Types XSD. Thirdly, the Bindings WSDL defines the Bindings used by this Producer. It imports the Interfaces WSDL and must include the standard SOAP HTTP(S) binding.

Both the Bindings and the PortTypes adhere to the following naming schema:

```
SPEC_VERSION_FACTOR_WSDLTYPE[_TYPESPECIFIC]
```

Where

```
SPEC           = WSRP
VERSION        = v1
FACTOR         = ServiceDescription | Markup | Registration
                   | PortletManagement
WSDLTYPE       = Binding | PortType
```

```
TYPESPECIFIC  = SOAP or …
```

Note that, additional standard Bindings may be added in future which define SOAP attachments mechanisms such as SwA/MIME or DIME. Such additional Bindings will add new `TYPESPECIFIC` binding names.

Each Producer service has to provide its own *Service Implementation WSDL* which contains a Service definition containing Ports for the various Bindings. This WSDL MUST import the WSRP *Binding WSDL*. The WSRP Bindings and Interfaces WSDLs are published on the OASIS web site (add reference to link here).

4.4 Using WSDLs in UDDI

In the domain of UDDI specifications, three documents can be identified which describe how WSDLs and their contents can be published to UDDI registries.
The best practices document “Using WSDL in a UDDI Registry, Version 1.08” ([add ref here]) clarifies the relationship between WSDL and UDDI and describes how WSDL can be used to help create UDDI business service descriptions.
The technical note “Using WSDL in a UDDI Registry, Version 2.0” describes a more sophisticated mapping of WSDL elements and UDDI data structures. This enables more specific and flexible UDDI queries based on WSDL artifacts and metadata. In this case, the UDDI registry becomes the authoritative source for the Web Service description in contrast to the WSDL file, which serves as the authoritative source in the prior document. However the V2 technical note maintains compatibility to the V1 best practices document and also defines a means to publish external Implementation WSDLs as the authoritative source.
Furthermore the UDDI V3 specification (see [add ref]) introduces another means to publish an external Implementation WSDL holding all necessary binding information, including the actual access points of a Web Service.

4.5 UDDI Versions Considerations

We base our recommendation on version 2 of the UDDI specifications for now, however, we will want to talk about V3 keys and V3 WSDL methodology in this document, too. The basic model is independent of these V2 / V3 considerations.

5 Publishing WSRP Services to UDDI

Refer to Alan’s abstract model here.

Compared to many Web Services scenarios, WSRP services prove more complex. Firstly, the WSRP Producer can be considered as a Web Service on its own, exposing multiple Bindings and PortTypes. One can view the WSRP Producer as the actual Web Service described through the WSRP WSDLs.

Secondly, Portlets can also be understood to be services. In contrast to Producers, Portlets are not full services in a Web Service sense. They do not explicitly expose PortTypes, Bindings or Access Points. Rather, a Portlet is exposed by its Producer. The Consumer interacts indirectly with Portlets through the Producer's infrastructure. The WSRP Portlet is addressed by a Portlet Handle defined in the Producer's scope.
In addition to the WSDL interface specification, WSRP services carry metadata which describes both the Producer and its Portlets. Prior to using a Producer or its Portlets the metadata has to be obtained.

Describe our decision for the Service Impl. WSDL as the authoritative source, to optionally publish Portlets etc.

Goals:

1. Publish Producer's WSDL to UDDI as the authoritative source.
   This means the actual access points need to be obtained from the WSDL. Second, metadata need to be obtained via the Service Description PortType.
2. Allow Consumers to search for WSRP Producers.
3. Optionally publish Portlets to UDDI with reference to their Producer.
4. Allow Consumer to search for WSRP Portlets, find the Producer (via Reference), obtain metadata,
5. Stay compatible with UDDI technote V1
6. Stay open towards UDDI technote V2

5.1 Overview

UDDI allows services to be published in the context of a business entity, as businessServices. Here we describe how WSRP Producers and Portlets may be published as UDDI businessServices.

Show Producers and Portlets published as businessServices, Portlets having a link to the Producer.

5.2 Publishing Producers

In essence Producers are Web Services. They expose PortTypes and Bindings a Consumer can interact with. These PortTypes and their Bindings are described in the WSRP WSDLs (see [add reference here]). Thus publishing a Producer corresponds very simply to the WSDL to UDDI mapping described by the UDDI technical notes. (refer to V1 and probably V2, Appendix A).
Given the published technical notes by the OASIS UDDI TC, it is anticipated that WSRP SHOULD NOT define its own WSDL to UDDI mapping but rather rely on the expertise of the OASIS UDDI TC on how to achieve the mapping.

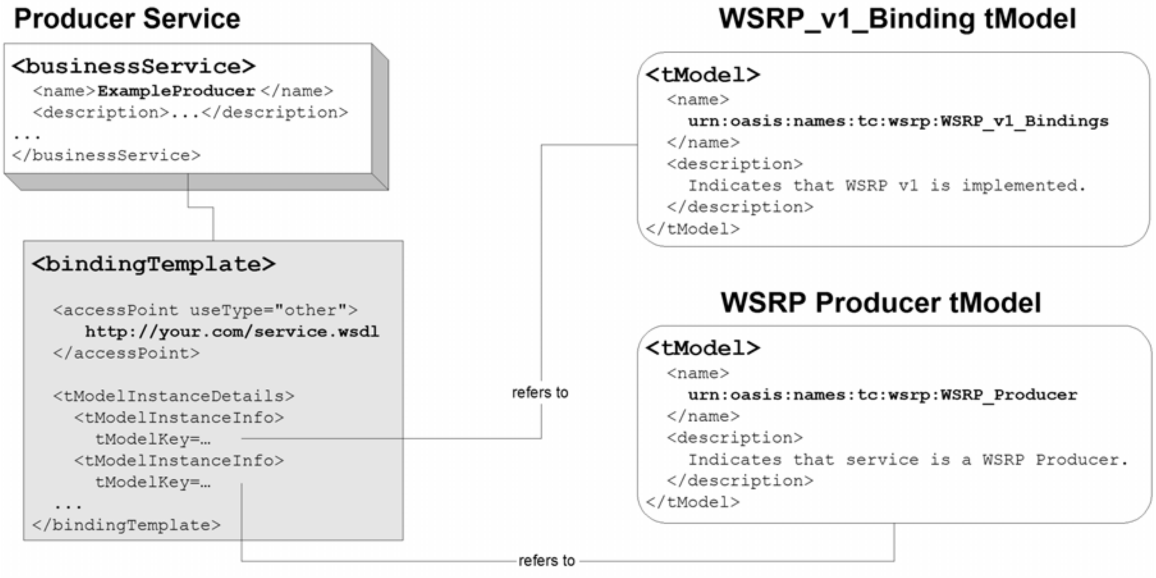
Generally, to publish a Producer the following steps need to be taken:

1. Publish the WSRP v1 Binding WSDL as a tModel. This tModel represents the technical fingerprint of the WSRP protocol.

2. Publish a “WSRP Producer” tModel. This tModel marks the businessService as being a WSRP Producer to allow for easy searching within the registry.
3. Publish the address of the Service Implementation WSDL as the authoritative source containing the actual service endpoints in a businessService representing the Producer (under some UDDI business).

Figure 1: Publishing Producers below shows the UDDI data structures used to publish WSRP Producers to a registry. Please note that the figure shows only the relevant elements in the structures to visualize the scheme used.

Basically, Producers are published as businessServices below the businessEntity structure (not shown here). The businessService structure uses a bindingTemplate to provide the required binding information. The access point element of the Producer bindingTemplate points to the URL of the *Service Implementation WSDL* providing the actual service endpoints. The bindingTemplate refers to two tModels. The first is indicating that the Producer at least implements the mandatory WSRP v1 PortTypes (it may implement other portTypes as well) and provides the mandatory WSRP v1 Bindings. Furthermore the WSRP Producer tModel is used to tag the Producer service as a WSRP Producer.



```
<businessService>
  <name>ExampleProducer </name>
  <description>...</description>
...
</businessService>
```

```
<bindingTemplate>

  <accessPoint useType="other">
     http://your.com/service.wsdl
  </accessPoint>

  <tModelInstanceDetails>
    <tModelInstanceInfo>
      tModelKey=…
    <tModelInstanceInfo>
      tModelKey=…
  ...
</bindingTemplate>
```

```
<tModel>
  <name>
    urn:oasis:names:tc:wsrp:WSRP_v1_Bindings
  </name>
  <description>
    Indicates that WSRP v1 is implemented.
  </description>
</tModel>
```

```
<tModel>
  <name>
    urn:oasis:names:tc:wsrp:WSRP_Producer
  </name>
  <description>
    Indicates that service is a WSRP Producer.
  </description>
</tModel>
```

Figure 1: Publishing Producers

[AK: diagram has WSRP_v1_Binding not WSRP_v1_Bindings]

5.2.1 Producer businessService

A WSRP Producer MUST be published using a UDDI businessService structure. Since there is no name or description of a Producer carried in its metadata, no explicit mapping between WSRP and UDDI need take place.

```
<businessService
```

```
    serviceKey="${SERVICEKEY_THIS_PRODUCER}"
    businessKey="${BUSINESSKEY_THIS_SERVICE_BELONGS_TO}">
  <name xml:lang="en">YourCo Producer</name>
  <description xml:lang="en">This is the WSRP Producer hosted by YourCo. This Producer
    offers some cool WSRP demo Portlets.
  </description>
  <bindingTemplates>
    …
  </bindingTemplates>
  <categoryBag>
    …
  </categoryBag>
</businessService>
```

[ak: a lot of the XML fragments could have an extra "…" before the last closing tag to indicate that they may carry extra subelements. Do we want to do this?]

Fields:

- `serviceKey`: The key value identifying this businessService entity.
- `businessKey`: The key of the business entity that is the parent of this businessService.
- `name`: A human readable name of the Producer accompanied by a unique xml:lang value. At least one name element SHOULD be provided.
- `description`: A language locale qualified text describing the Producer. This field is optional.
- `bindingTemplates`: This field MUST contain a bindingTemplate structure as defined in section [ref here]. It MAY contain further bindingTemplates relevant to this Producer. The next section describes this structure in detail.
- `categoryBag`: The field MAY contain several name-value-pairs used to tag the Producer with specific taxonomy information.

5.2.2 Producer bindingTemplate

UDDI bindingTemplates are used to provide technical descriptions of Web Service instances. For WSPR, the Producer service's bindingTemplate is used to provide the URL of the WSDL holding the actual access points of the producer. It also refers to the WSRP v1 Bindings WSDL and indicates that the published businessService is a WSRP Producer.

```
<bindingTemplate
  bindingKey="${BINDINGKEY_OF_THIS_BINDINGTEMPLATE}"
  serviceKey="${SERVICEKEY_OF_THIS_PRODUCER}">
  <description xml:lang="en">Description of this template</description>

  <accessPoint useType="other">
    http://yourco.com/producer/sevice.wsdl
  </accessPoint>

  <tModelInstanceDetails>
```

```
    <tModelInstanceInfo
        tModelKey="${TMODELKEY_WSRP_V1_BINDINGS}">
    </tModelInstanceInfo>
    <tModelInstanceInfo
        tModelKey="${TMODELKEY_WSRP_PRODUCER}">
    </tModelInstanceInfo>
  </tModelInstanceDetails>
</bindingTemplate>
```

Fields:

- bindingKey: Key value identifying this bindingTemplate.
- serviceKey: The key of the businessService that contains this bindingTemplate.
- description: A language locale qualified text describing the usage of this bindingTemplate. This field is optional.
- accessPoint: The bindingTemplate MUST contain this field. The accessPoint MUST hold the URL of the http(s) resource providing the authoritative Service Implementation WSDL of the Producer. The Service Implementation WSDL MUST contain only one wsdl:service element. The wsdl:service MUST contain port definitions for the mandatory WSRP V1 PortTypes and Bindings.
- tModelInstanceDetails: This container MUST contain at least contain two tModelInstanceInfo structures. One of these MUST refer to the WSRP_v1_Bindings tModel defined in section [ref here]. A second MUST refer to the WSRP_PRODUCER tModel defined in section [ref here].

5.2.3 WSRP_v1_Bindings tModel

This tModel is used to indicate conformance to the WSRP v1 specification. A bindingTemplate that refers to this tModel states that the Web Service is implementing at least the mandatory WSRP PortTypes and providing the default SOAP Bindings for these PortTypes.

For the UBR (Universal Business registry), the tModel will be published by the WSRP TC.

```
<tModel tModelKey="${TMODELKEY_WSRP_V1_BINDINGS}">
  <name>
    urn:oasis:names:tc:wsrp:WSRP_v1_Bindings
  </name>
  <description xml:lang=”en”>
    The purpose of this tModel is to …
  </description>
  <overviewDoc>
    <overviewURL>
        http://www.oasis-open.org/committees/wsrp/specifications/version1/wsrp_v1_bindings.wsdl
    </overviewURL>
  </overviewDoc>

  <categoryBag>
    <keyedReference
        tModelKey="uuid:C1ACF26D-9672-4404-9D70-39B756E62AB4"
        keyName="uddi-org:types"
        keyValue="wsdlSpec"/>
  </categoryBag>
```

```
</tModel>
```

Fields:

- tModelKey: The key value identifying the tModel.
- name: The tModel name. Note that, the name MUST NOT have a language attribute.
- description: A language qualified description of this tModel.
- overviewURL: The URL MUST hold the location of the WSRP v1 Bindings WSDL.
- categoryBag: The categoryBag MUST contain a keyedReference to the uddi-org:types taxonomy tModel. The keyValue of this keyedReference MUST be "wsdlSpec".

5.2.4 WSRP Producer canonical tModel

In addition to the tModel above, the WSRP Producer tModel requires tagging the Producer's businessService as being of type "WSRP Producer". This tagging provides an easy way to find all/any WSRP Producers in the registry. The WSRP Producer tModel will be published to the UBR by the WSRP TC.

```
<tModel tModelKey="${TMODELKEY_WSRP_PRODUCER}">
  <name>
    urn:oasis:names:tc:wsrp:WSRP_PRODUCER
  </name>
  <description xml:lang="en">
    The purpose of this tModel is to …
  </description>
  <overviewDoc>
    <overviewURL>
        URL to this note.[TO DO]
    </overviewURL>
  </overviewDoc>
</tModel>
```

Fields:

- `tModelKey`: The key value identifying the tModel.
- `name`: The tModel name. Note that, the name MUST NOT have a language attribute.
- `description`: A language locale qualified description of this tModel.
- `overviewURL`: The URL MUST hold the location of this note on the OASIS web site.

5.2.5 Producer's Metadata

Producer metadata is not directly published to UDDI. The metadata MUST be retrieved using the ServiceDescription PortType published by the Producer.
Currently there is no desire to duplicate this metadata by also publishing it to registries. However, in future, use-cases might come up which a desire to search for Producers

defining certain metadata values in UDDI and which would require publishing Producer metadata to UDDI.
Such considerations are deferred to later versions of this document.

5.3 Publishing Portlets

By publishing Producer there is already a means for Consumers to discover the Portlets a Producer offers outside UDDI, by using the Producer’s Service Description PortType. However, to enhance the end-user experience and allow a direct search for Portlets in registries the publisher MAY additionally publish one or more Portlets (we recommend that all portlets are published to UDDI).

Portlets are different from Producer services. They don’t expose interfaces, access points or bindings directly. A Consumer interacts with Portlets indirectly via the Producer, addressing them by their Portlet Handles. Therefore the normal WSDL to UDDI mappings do not match our requirements here. Portlets should not be published as Web Services by re-exposing the Producer’s access point and bindings. Instead Portlets should reference the Producer that manages them.
The following sections describe how the UDDI data structures MUST be utilized to publish Producer offered Portlets defined by WSRP.

[AK: I still have some concerns about requiring this. I would rather allow the “producer reference” to take one of several forms: service key, federation link, wsdl url. This would be included either as a keyedReference or as a second bindingTemplate on the service.]

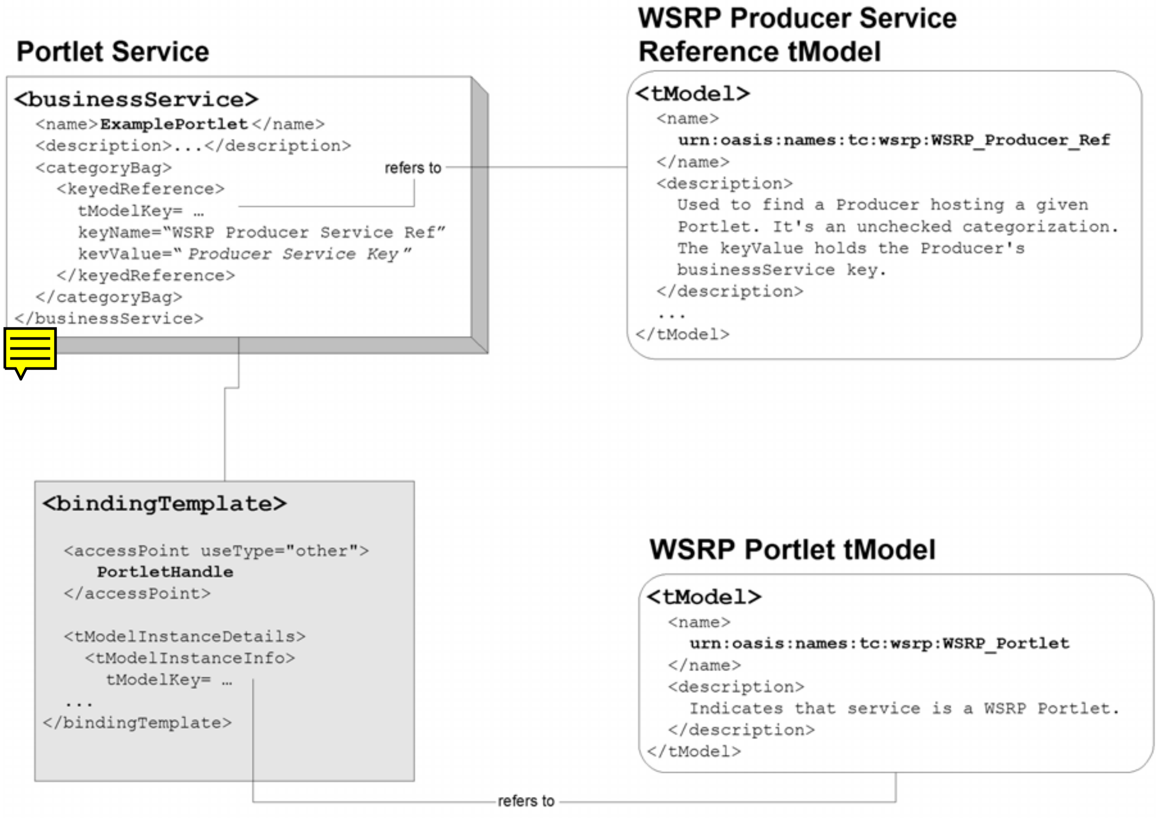
The following steps need to be taken to publish a Portlet:

1. Publish the “WSRP Portlet” tModel. This tModel marks a businessService as being a WSRP Portlet to allow an easy search within the registry.
2. Publish a “WSRP Producer Service Reference” tModel. This categorization tModel can be used as a means to model the relationship between Portlets and hosting Producers.
3. Publish the Portlet Handle to the registry, in a portlet businessService, and
4. Add a reference to the hosting Producer’s businessService when publishing the Portlet as a businessService

[AK: need to order steps wrt UDDI update requests so businessService is created last.]

Figure 1: Publishing Producers below shows the UDDI data structures used to publish WSRP Portlets to a registry. Please note that the figure shows only the relevant elements in the structures to visualize the scheme used.
Basically, Portlets are published as businessServices below the businessEntity structure (not shown here). The businessService structure uses a bindingTemplate to provide the required binding information. The access point element of the Portlet bindingTemplate holds the Portlet Handle. The bindingTemplate refers to the WSRP Portlet tModel which tags the businessService as a WSRP Portlet. The Portlet businessService uses a categorization scheme to add a reference to the hosting Producer. The keyedReference refers to the WSRP Producer Service Reference tModel. The keyValue of the reference

stores the Producer’s businessService key. Using a categorization scheme to represent the relationship between Producers and Portlets allows searches for Portlet hosted by a particular Producers.



```
<businessService>
  <name>ExamplePortlet</name>
  <description>...</description>
  <categoryBag>
    <keyedReference>
      tModelKey= …
      keyName=“WSRP Producer Service Ref”
      kevValue=“ Producer Service Key ”
    </keyedReference>
  </categoryBag>
</businessService>
```

```
<tModel>
  <name>
    urn:oasis:names:tc:wsrp:WSRP_Producer_Ref
  </name>
  <description>
    Used to find a Producer hosting a given
    Portlet. It's an unchecked categorization.
    The keyValue holds the Producer's
    businessService key.
  </description>
  ...
</tModel>
```

```
<bindingTemplate>

  <accessPoint useType="other">
     PortletHandle
  </accessPoint>

  <tModelInstanceDetails>
    <tModelInstanceInfo>
      tModelKey= …
  ...
</bindingTemplate>
```

```
<tModel>
  <name>
    urn:oasis:names:tc:wsrp:WSRP_Portlet
  </name>
  <description>
    Indicates that service is a WSRP Portlet.
  </description>
</tModel>
```

Figure 2: Publishing Portlets

5.3.1 Portlet businessService

A WSRP Portlet service MUST be published using a UDDI businessService structure. In Order to retrieve the Producer, hosting the Portlet, a reference to the Producer service MUST be set up. The tModel “WSRP Producer Service Reference” provides a means to establish this reference.

Although the authoritative source of the Portlet metadata is the WSRP PortletDescription structure, which must be obtained through the Producer’s Service Description PortType, it will be advantageous to allow the mapping of some of the fields from WSRP’s PortetDescription structure to appropriate fields in the businessService structure for query / search based discovery.

```
<businessService
     serviceKey="${SERVICEKEY_OF_THIS_PORTLET}"
     businessKey="${BUSINESSKEY_THIS_SERVICE_BELONGS_TO}">
  <name xml:lang=”en”>StockQuote Portlet</name>
  <description xml:lang=”en”>This Portlet is the one and Portlet displaying  stock quotes.
  </description>
  <bindingTemplates>
```

```
    …
  </bindingTemplates>
  <categoryBag>
    <keyedReference
        tModelKey=”${TMODELKEY_WSRP_PRODUCER_SERVICE_REFERENCE}”
        keyName=”WSRP Producer Service Refererence”
        keyValue=”${SERVICEKEY_PRODUCER}”
    />
  </categoryBag>
</businessService>
```

Fields:

- `serviceKey`: The key value identifying this businessService entity.
- `businessKey`: The key of the business entity that contains this businessService.
- `name`: A human readable name of the Portlet accompanied by a non-duplicated xml:lang value. At least one name element SHOULD be provided.
  The name field with the correct localized language attribute SHOULD be mapped to the title field of the WSRP PortletDescription structure.
- `description`: A language qualified text describing the Portlet. This field is optional. The description element with the matching localized language attribute SHOULD be mapped to the description field of the WSRP PortletDescription structure.
- `bindingTemplates`: This field MUST contain a bindingTemplate structure as defined in the next section. It MAY contain further bindingTemplates relevant to this Portlet.

`categoryBag`: This field MAY contain a keyed reference to the Producer’s businessService entity in the same UDDI registry. The keyed reference tModelKey attribute MUST hold the tModelKey value of the WSRP Producer Service Reference tModel. The value of the keyName attribute MUST be “WSRP Producer Service Reference”. Finally, the keyValue must store the serviceKey value of the businessService entity of the Producer hosting that particular Portlet.
The categoryBag MAY contain further entries.

5.3.2 Portlet bindingTemplate

UDDI bindingTemplates are used to provide technical descriptions of Web Service instances. For WSRP, the Portlet bindingTemplate is used to provide the Portlet Handle and to indicate that the published businessService is a WSRP Portlet.

```
<bindingTemplate
  bindingKey="${BINDINGKEY_OF_THIS_BINDINGTEMPLATE}”
  serviceKey="${SERVICEKEY_OF_THIS_PORTLET}">
  <description xml:lang=”en”>Description of this template</description>

  <accessPoint useType="other">
    StockQuote_Handle_0123456789
  </accessPoint>

  <tModelInstanceDetails>
```

```
    <tModelInstanceInfo
        tModelKey="${TMODELKEY_WSRP_PORTLET}">
    </tModelInstanceInfo>
  </tModelInstanceDetails>
</bindingTemplate>
```

Fields:

- bindingKey: Key value identifying this bindingTemplate.
- serviceKey: The key of the businessService contains this bindingTemplate.
- description: A language locale qualified text describing the usage of this bindingTemplate. This field is optional.
- accessPoint: The bindingTemplate MUST contain this field. The accessPoint MUST hold the Portlet Handle as defined by the handle field of the WSRP PortletDescription structure for the Portlet. The type of the accessPoint MUST be “other”.
- tModelInstanceDetails: This container MUST at least contain one tModelInstanceDetails structure that references the WSRP_PORTLET tModel defined in section [ref here].

5.3.3 WSRP Producer Service Reference canonical tModel

The WSRP Producer Service Reference tModel provides the means to express the relationship of a Portlet businessService to a Producer businessService. Using this categorization scheme one can find a Producer businessService entry representing the Producer hosting this particular Portlet. With this information, one can retrieve all necessary data to address the Portlet.
This tModel is published by the WSRP TC into the UBR.

```
<tModel tModelKey="${TMODELKEY_WSRP_PRODUCER_REFERENCE}" >
  <name>WSRP_v1_ProducerReference</name>
  <description xml:lang=”eng”
  This tModel is a taxonomy tModel used to identify a relationship to a Producer’s
  businessService UDDI entity
  </description>
  <overviewDoc>
    <overviewURL>
       URL to this note.
    </overviewURL>
</overviewDoc>
<categoryBag>
    <keyedReference
        tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
        keyValue="unchecked"
    />
    <keyedReference
        tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4" [ak: uppercase :C1..B4” in
spec
        keyValue="categorization"
    />
</categoryBag>
</tModel>
```

Fields:

- `tModelKey`: The key value identifying the tModel.
- `name`: The tModel name. Note that, the name MUST NOT be accompanied with a language attribute.
- `description`: A language locale qualified description of this tModel.
- `overviewURL`: The URL MUST hold the location of this note on the OASIS web site.

[AK: Should the “categorization” TModel be an “identifier” one instead? Both seem not to be a complete fit. Do we have any argument as to why “categorization” is better?]

5.3.4 WSRP Portlet canonical tModel

The WSRP Producer tModel tags the Portlets’s businessService as being of type “WSRP Portlet”. This tagging provides an easy way to find WSRP Portlets in the registry. The WSRP Portlet tModel will be published to the UBR by the WSRP TC.

```
<tModel tModelKey="${TMODELKEY_WSRP_PORTLET}">
   <name>
      urn:oasis:names:tc:wsrp:WSRP_PORTLET
   </name>
   <description xml:lang=”en”>
      The purpose of this tModel is to …
   </description>
   <overviewDoc>
      <overviewURL>
            URL to this note.
      </overviewURL>
   </overviewDoc>
</tModel>
```

Fields:

- `tModelKey`: The key value identifying the tModel.
- `name`: The tModel name. Note that, the name MUST NOT be accompanied with a language attribute.
- `description`: A language locale qualified description of this tModel.
- `overviewURL`: The URL MUST hold the location of this note on the OASIS web site.

5.3.5 Portlet’s Metadata

Besides the Portlet Handle, the Portlet title and textual description, all further Portlet metadata is not directly published to UDDI. All remaining metadata MUST be retrieved using the ServiceDescription PortType exposed by the Producer.
Currently there is no need foreseen to require publishing this metadata. However, in future, use-cases might arise which require searching for Portlets defining certain metadata values in UDDI and thus need additional mapping to UDDI.
This will be deferred to later versions of this document.

6 Private UDDI Registries

Do we want to mention the caveats here? The handling blablabla…

7 References

7.1 Normative

[RFC2119] *S. Bradner, Key words for use in RFCs to Indicate Requirement Levels, IETF RFC 2119, March 1997. Available at* http://www.ietf.org/rfc/rfc2119.txt.

[1] *WSRP specification, current draft. Available at* http://oasis-open.org/committees/wsrp/documents/wsia_wsrp_interface_spec_v0.85.pdf

[2] *Using WSDL in a UDDI registry, Version 2.0. Currently writing draft, available mid-february 2003.*

[3] *Using WSDL in a UDDI Registry 1.08. Available at* http://uddi.org/pubs/wsdlbestpractices.pdf

[4] *Web Services Description Language (WSDL) 1.1, March 15, 2000. Available at* http://www.w3.org/TR/wsdl

[5] *UDDI Version 2.03 Data Structure Reference, July 7, 2002. Available at* http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.pdf.

[6] *UDDI Version 3.0 Published Specification, 19 July 2002. Available at* http://www.uddi.org/pubs/uddi-v3.00-published-20020719.pdf.

[7] *URL of WSRP Bindings WSDL*

[7] *URL of WSRP Interfaces WSDL*

[7] *URL of WSRP Types XSD*

Appendix A. Revision History

Rev	Date	By Whom	What
draft-01	09/12/03	Richard Jacob	First draft
draft-02	09/18/03	Richard Jacob	Added publishing of Portlets
draft-03	01/16/04	Andre Kramer	Editorial suggestions, added Producer Servcie Reference Binding Template
draft-04	01/20/04	Richard Jacob	added figures and descriptions for Producer and Portlet publishing
draft-05	02/12/04	Andre Kramer, Richard Jacob	editorial changes, added line numbering