Web Services Resource Framework (WSRF) – Primer v1.2

Committee Draft 02 - 23 May 2006

Document identifier: wsrf-primer-1.2-primer-cd-02

Location:

http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-02.pdf

Author & Editor:

Tim Banks, IBM <Tim_Banks@uk.ibm.com>

Abstract:

Service interfaces often imply the need for some form of persistent interaction with the clients of the service. This may be apparent in a conversational style of use of a particular service interface in which some aspect of the result of one operation influences the execution of the succeeding ones.

The purpose of the Web Services Resource Framework (WSRF) is to define a generic framework for modelling and accessing persistent resources using Web services so that the definition and implementation of a service and the integration and management of multiple services is made easier.

This document provides an introduction to WSRF using accessible examples.

Status:

This document is published by this TC as a committee draft. Committee members should send comments on this document to the wsrf@lists.oasis-open.org list. Others may submit comments to the TC via the web form found on the TC's web page at http://www.oasis-open.org/committees/wsrf. Click the button for "Send A Comment" at the top of the page. Submitted comments (for this work as well as other works of the TC) are publicly archived and can be viewed at http://lists.oasis-open.org/archives/wsrf-comment/.

Contents

Figures and Tables

1. Sample XML Files and Namespaces

Extracts of the XML files which define services are included in the text but, for the sake of clarity, some details are omitted. The complete versions of the examples can be found via the references in section 14 and via hyperlinks within the examples in the electronic version.

XML namespace declarations are omitted in the text examples. However, the following namespace prefixes are used throughout and in the associated WSDL and XSD files which define elements in the namespaces. The following convention is used to make the origin of the XML definitions clearer:

- Prefixes beginning “wsrf-” refer to XML elements defined by the WSRF specifications.
- The following prefixes are associated with the examples:
  - Prefixes “ssc”, “sscw” and “ws-ssc” belong to the SimpleShoppingCart
  - Prefixes “pr”, “prw” and “ws-pr” belong to the Printer
  - Prefix “sc” and “scw” belongs to the ShoppingCart
  - Prefix “prjg” and “prjgw” belong to the PrinterAndJobGroup

- Other namespaces belong to standard Web service and XML specifications on which WSRF depends.

Namespaces are contained in the following table. The shaded rows are the namespaces defined for the examples used in this Primer. Other namespaces are defined by the WSRF specifications, or standards on which WSRF depends.

Prefix	Namespace
pr	Schema definition for the Printer Resource Properties document
prjg	Schema definition for the Printer and Job Group Resource Properties document
prw	Printer WSDL message definitions
rpimpl	Schema definition for implementation-dependent WS-Addressing reference parameters.
sc	Schema definition for the ShoppingCart
ssc	Schema definition the WSRF form of SimpleShoppingCart
sscw	WSDL message definitions in WSRF form of the SimpleShoppingCart
scw	WSDL message definitions in WSRF form of the ShoppingCart
ws-pr	Message definitions for the non-WSRF Web service version of the Printer.
ws-prw	WSDL definitions for the non-WSRF Web service version of the Printer
ws-ssc	schema definition of the non-WSRF Web service form of the SimpleShoppingCart
ws-sscw	WSDL message definitions in non-WSRF Web service form of the SimpleShoppingCart
soap	http://schemas.xmlsoap.org/wsdl/soap/ [SOAP]
soap-env	http://schemas.xmlsoap.org/soap/envelope/ [SOAP]
wsa	http://www.w3.org/2005/08/addressing [WS-Addressing]
wsaw	http://www.w3.org/2005/02/addressing/wsdl [WS-Addressing]
wsdl	http://schemas.xmlsoap.org/wsdl/
wsnt	http://doc.oasis-open.org/wsn/b-2 [WS-BaseNotification]
wsrf-rp	http://docs.oasis-open.org/wsrf/rp-2 - WSRF ResourceProperties Schema
wsrf-rpw	http://docs.oasis-open.org/wsrf/rpw-2 - WSRF ResourceProperties WSDL
wsrf-bf	http://docs.oasis-open.org/wsrf/bf-2 - WSRF BaseFaults Schema
wsrf-bfw	http://docs.oasis-open.org/wsrf/bfw-2 - WSRF BaseFaults WSDL
wsrf-rl	http://docs.oasis-open.org/wsrf/rl-2 - WSRF ResourceLifetime schema

wsrf-rlw	http://docs.oasis-open.org/wsrf/rlw-2 - WSRF ResourceLifetime WSDL
xsd	http://www.w3.org/2001/XMLSchema
xsi	http://www.w3.org/2001/XMLSchema-instance

2. About this Primer

This primer is an approachable introduction to WSRF. You should read it if you need to get a summary of the facilities offered. These are presented using simple, familiar application examples, namely a shopping cart and a print server.

For more detailed information and advice about the practical implementation and particular topics, you may find the companion Application Notes [AppNotes] document useful. For questions which need complete, definitive answers, you should look at the specifications themselves: these are referenced in the text where appropriate and section 13 ('References') contains their URLs.

The examples are designed to be understood without prior experience but, to get the full benefit from this Primer, you will need to be familiar with XML [XML Primer] and the concept of the XML schema language used to define XML documents since WSRF uses XML documents to describe the resources. We also assume you are familiar with Web services and with the WSDL [WSDL] description and SOAP message dialects that are used to define the messages that clients use to communicate with Web services. Section 14 contains a pointer to the file that contains the XML documents which describe the services used in the examples. Fragments of these files are included in this document, and to make the text more readable, namespace declarations are omitted: the prefixes and corresponding namespaces are described in section 1. The outline of the rest of this document is as follows:

- Section 3 explains WSRF in a nutshell.
- Section 4 introduces the examples and is an overview of the operations defined by WSRF.
- Sections 5 and 6 use a simple shopping service to explain how WS-Resources are defined and the way in which the definitions are used in the Get- and PutResourcePropertyDocument operations.
- Section 7 explains the definition of Web service faults using WSRF.
- Section 8 develops an example based on a networked Printer and the operations for manipulating properties of the printer. This example uses more detailed operations (provided by WSRF) than are required in the simple shopping service.
- Section 9 shows an alternative shopping service with a more complex document structure, and also introduces the WS-ResourceLifetime specification.
- Section 10 develops the Printer example further to show how print jobs are represented using the mechanisms introduced in section 9.
- Section 11 explains how a directory of printers and print jobs can be constructed using the WS-ServiceGroup specification, and the use of Notification messages that can be generated by WS-Resources to inform interested parties about changes of state.
- Section 12 is a Glossary of Terms.

3. WSRF in a Nutshell

A Web service is characterized by the messages that flow to and from it, and which are defined in a Web Services Description Language [WSDL] document. Any resource manipulated by the service, such as a shopping cart modeled in an internet shopping site, or a printer which represents part of a remote printing service, or a print job created by that service, needs to be identified and described by the messages that are exchanged.

WSRF introduces the idea of an XML document description, called the Resource Properties document schema, which is referenced by the WSDL description of the service and which explicitly describes a view of the shopping cart, printer, print job, or whatever, with which the client interacts. A resource described in this way is called a WS-Resource. By exploiting the Resource Properties document schema, WSRF enables the mechanical definition of simple, generic messages which interact with the WS-Resource.

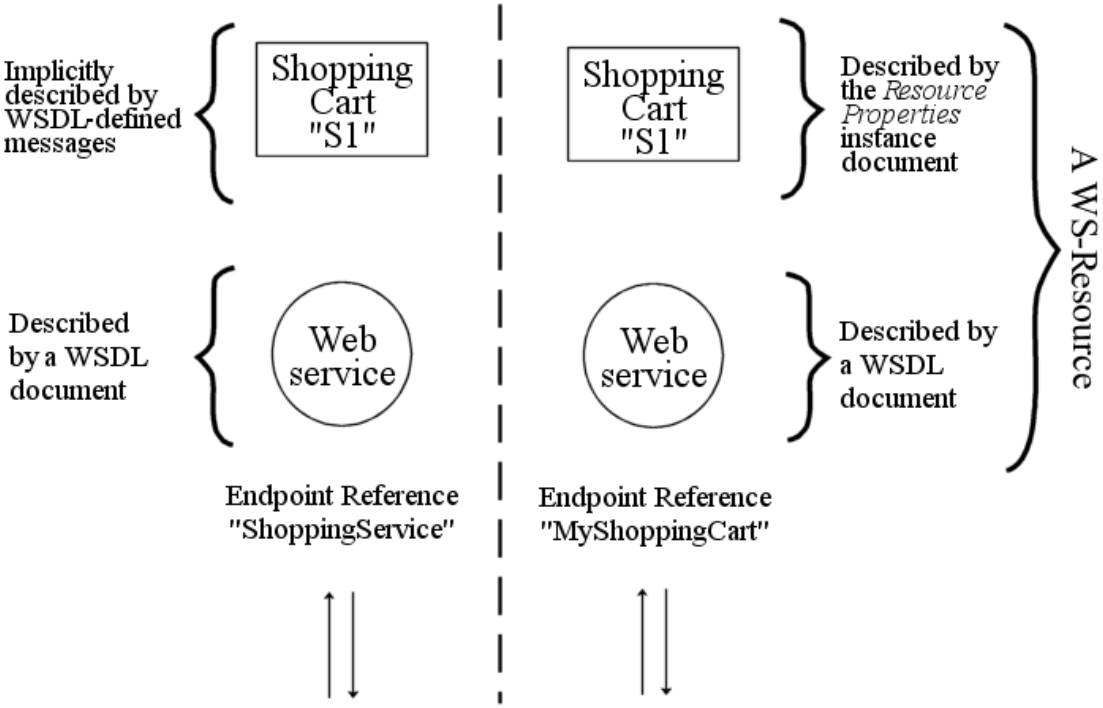

Figure 1: Non-WSRF Web services and WSRF

Figure 1 shows a comparison of a simple shopping service built with non-WSRF Web services (on the left) and WSRF (on the right): details of this example can be found in section 5. The left-hand side of the figure shows a ShoppingCart resource called 'S1' accessed through a Web service at a particular location. The location is described by a WS-Addressing [WS-Addressing] EndpointReference (EPR), which contains the URI of the Web service endpoint as follows:

```
<wsa:EndpointReference>
  <wsa:Address>http://www.example.com/WSSimpleShoppingService</wsa:Address>
</wsa:EndpointReference>
```

This EPR is known as "ShoppingService" by the requester.

EPRs and ports

The use of an EndpointReference (EPR) to describe the location of a Web service is a recent innovation in Web services architecture. Just like the port in the service element of WSDL, the EPR contains a URI [URI] which indicates the location of the Web service.

We describe Web services locations using EPRs because this technique provides the clearest comparison with the mechanisms introduced by WSRF.

Arrows in Figure 1 represent messages sent between the requester and the Web service. What is not shown is the mechanism by which the specific ShoppingCart 'S1' is identified amongst the many carts that may be accessed through the service at the same location: the ShoppingCart identifier must be present somewhere in the messages and is typically passed as a parameter to the Web service and described in the WSDL.

WSRF avoids the need to describe the identifier explicitly in the WSDL description. Instead, the identifier can be packaged as part of the EPR and implicitly included in all messages addressed through the EPR according to the rules of WS-Addressing. The resource identifier itself ceases to be of interest or concern to the requesting application which simply uses the EPR as a reference to a specific WS-Resource. The right-hand side of Figure 1 shows the same service in the WSRF view: a combined Web service and ShoppingCart described by a Resource Properties document which are together known as a WS-Resource and referenced by the requester as "MyShoppingCart". The EPR might look like this:

```
<wsa:EndpointReference>
  <wsa:Address>
    http://www.example.com/SimpleShoppingService
  </wsa:Address>
  <wsa:ReferenceParameters>
    <rpimpl:CartId>S1</rpimpl:CartId>
  </wsa:ReferenceParameters>
</wsa:EndpointReference>
```

The forms of the SOAP messages for the two forms of the simple shopping service are shown in Figure 2: non-WSRF Web services on the left and the WSRF version on the right.

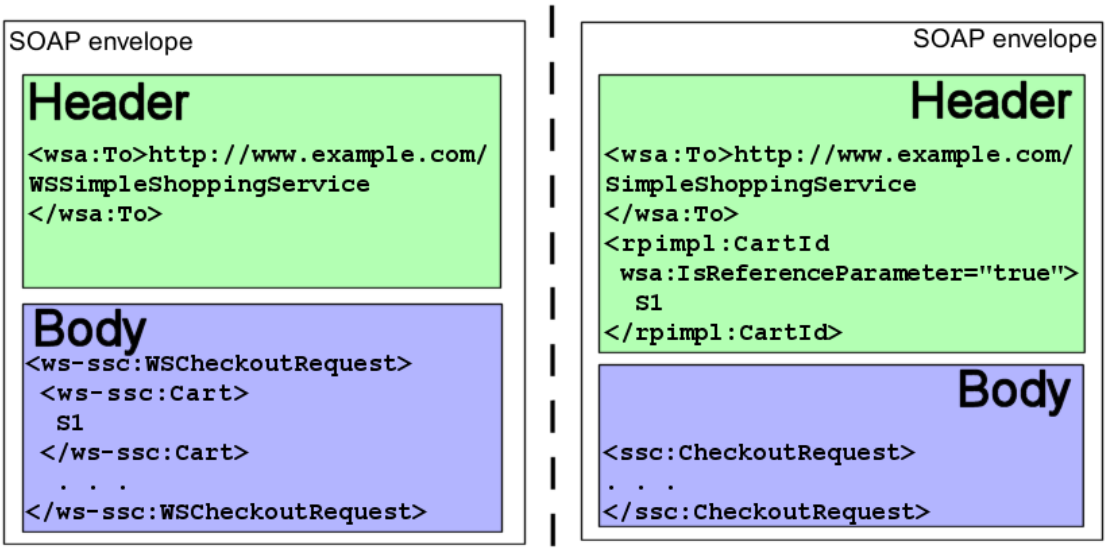

Figure 2: Equivalent SOAP messages for non-WSRF Web services and WSRF

By removing the resource identifier from the body of the messages, WSRF simplifies the form of the WSDL that describes the service. The WSDL for a message that, for example, adds an item to a cart need only be concerned about the item to be added; the requesting application is not responsible for maintaining the identity of the cart to which the item must be added.

WSRF exploits this simplification of the service definition to provide standardized forms for messages which interact with a WS-Resource, namely:

- A way to retrieve and replace the whole Resource Properties document describing the resource.
- Several ways to retrieve and replace named parts of the document.

These standardized messages not only simplify definition and implementation of services: the task of any client application which must deal with multiple resources is simplified if the resource definitions and operations follow the same pattern. In particular, this is advantageous to resource management systems.

WSRF also provides standard solutions to the following issues which commonly arise in distributed systems and whose description is facilitated by the use of the WS-Resource concept.

- A way to describe and control the lifetime of a WS-Resource.
- The systematic description and handling of fault messages.
- The aggregation of information about resources and services to form directories.
- The description of messages which may publish, to interested parties, changes in state of WS-Resources, including changes their property values.

4. About the Examples

The Primer uses four examples, each of which is based on a description of a shopping cart or a printer. All of the examples illustrate a characteristic typical for WSRF –

multiple resource instances accessed via a uniform interface. They also illustrate different styles of interaction and different ways to represent information and behaviour of the underlying resources.

The following subsections introduce the examples.

4.1. The WS-Resource Properties Document as a Single Unit

This example introduces the definition of the SimpleShoppingCart and demonstrates using a shopping service to create, retrieve and update the cart via the SimpleShoppingCart document.

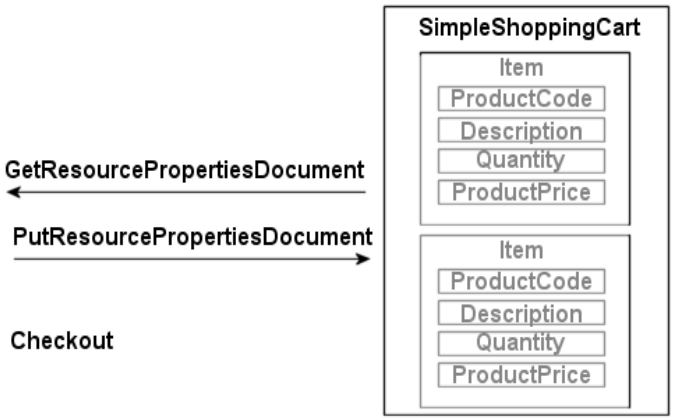

Figure 3: The SimpleShoppingCart

To make clear the facilities introduced by WSRF, the example illustrates how definitions and messages are constructed using WSDL and XML but without WSRF, and then how the same problem can be solved using WSRF.

The Get/PutResourcePropertyDocument operations used in the example are defined by the Web Services Resource Properties specification [WS-ResourceProperties]. The example is covered in sections 5 and 6.

4.2. A Collection of Properties

This example uses a printer as the example to show that a WS-Resource can represent both physical devices and logical entities. In this case, physical properties of a printer are presented via a Resource Properties document and are modified via operations on the document to illustrate the standard operations available in WSRF. These operations identify, retrieve and modify named properties of the printer, such as the printer state or the number of queued jobs.

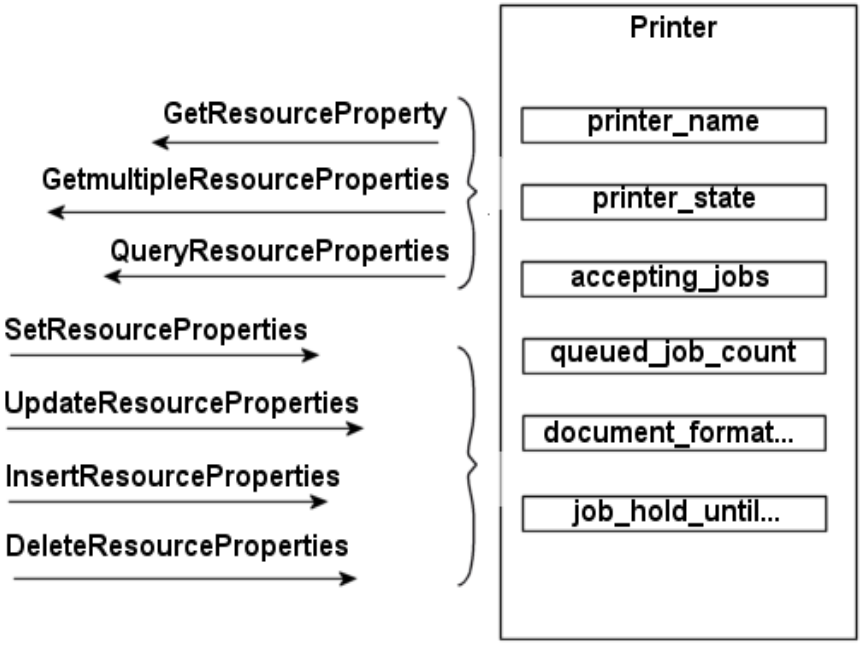

Figure 4: The Printer

This example also compares the message definitions needed in WSRF with the equivalent definitions for a service that does not use WSRF.

The construction of the logical jobs managed by a printer is not described in detail, but is identical to the construction of the items within the shopping cart illustrated in the next section. The printer example is developed further in section 4.4.

The operations used in the example are defined by the Web Services Resource Properties specification [WS-ResourceProperties]. The example is covered in section 8.

4.3. Accessing Parts of a Complex Resource

This example uses a modified form of the shopping cart which illustrates the selective retrieval and update of repetitive parts of the Resource Properties document by identifying and updating items of shopping in the cart. Each item within the shopping cart is itself an addressable WS-Resource contained within the shopping cart WS-Resource. Each Item WS-Resource has a Resource Properties document that is a fragment of the shopping cart. The example also illustrates WSRF's destroy operation which is used to remove items, and a facility provided by WSRF for a timeout on the cart if there is no checkout instruction from the requester.

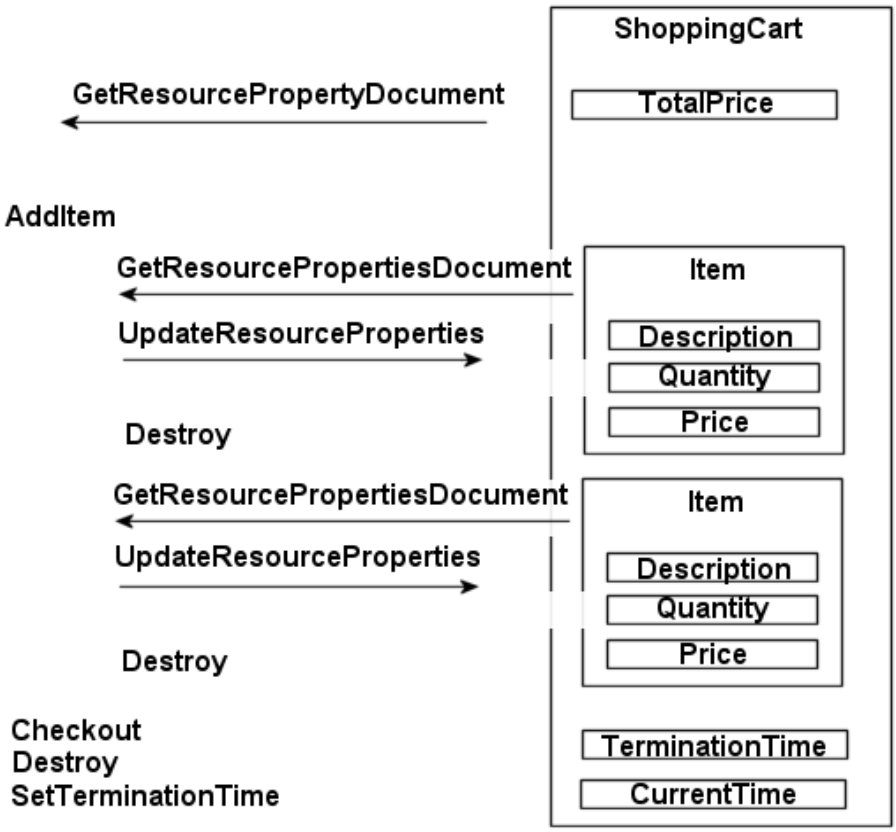

Figure 5: The Shopping Cart as multiple WS-Resources

The operations used in the example are defined by the Web Services Resource Properties specification [WS-ResourceProperties], except for the Destroy and SetTerminationTime operations which are defined by Web Services Resource Lifetime specification [WS-ResourceLifetime]. The AddItem and Checkout operationsare service-specific and are not defined by WSRF.

The example is covered in section 9.

4.4. Directories and other Groups of Services

This example shows how to use the WSRF ServiceGroup [WS-ServiceGroup] by constructing a directory which lists a collection of printers along with their status values and the print_jobs associated with them. The resulting collection is called the PrinterAndJobGroup. Query operations on the directory can be used, for example, to discover which printers have stopped working, or to list jobs on any of several printers which have been created by a particular user. The example also explains how the requirements for the construction of this kind of directory of dynamic information are somewhat different from the kind of collection illustrated in the shopping cart or the printer and its jobs:

The use of messages to update the directory information with the current status of printers and jobs is used as an illustration of a general facility to create notification messages as a result of changes in a WS-Resource's property values.

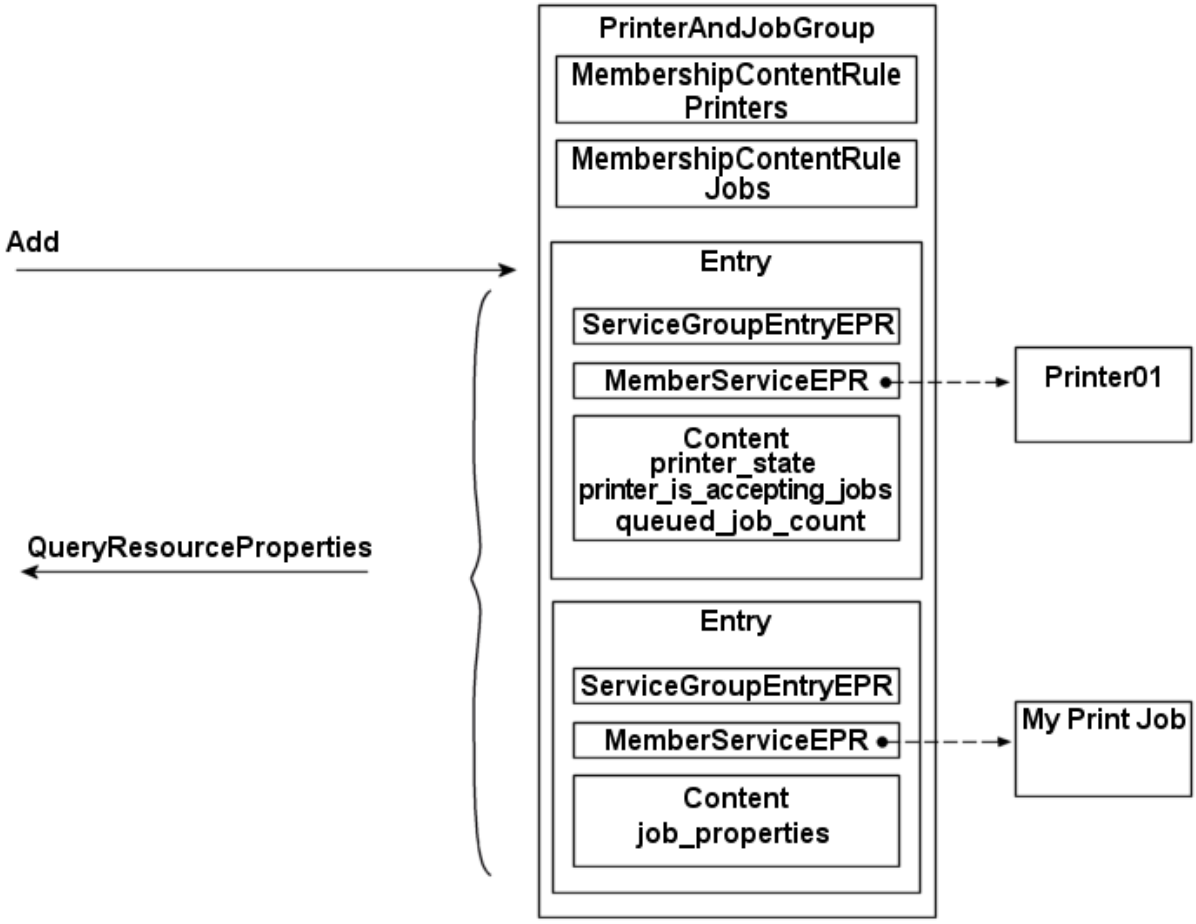

Figure 6: Printer and Job directory

This example is covered in section 11.

5. The Simple Shopping Service

This section illustrates WSRF service interactions using a shopping service which enables a user to create, retrieve and update the cart via "Get" and "Put" operations on the SimpleShoppingCart document.

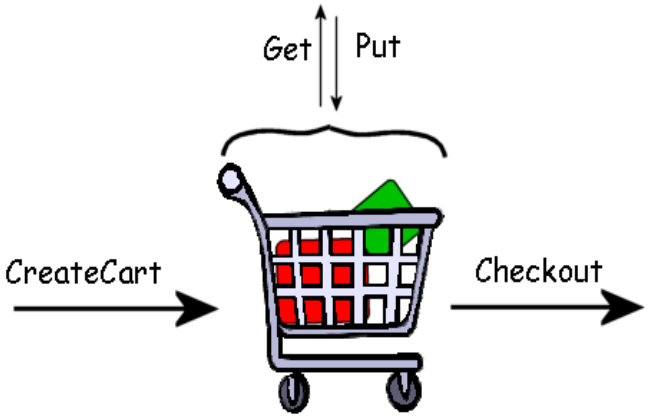

Figure 7: The simple shopping service

The shopping service provides access to a searchable product catalogue. A ProductCode and an order Quantity are then used to form the initial item in a shopping cart, to which additional products and quantities can be added. A Checkout operation causes the purchase interaction to start via a payment service and the contents of the cart then become the contents of a Customer Order. The cart is destroyed.

5.1. Using non-WSRF Web Services

5.1.1. The WSDL Description

The WSDL description for the WSSimpleShoppingService Web service is summarized below.

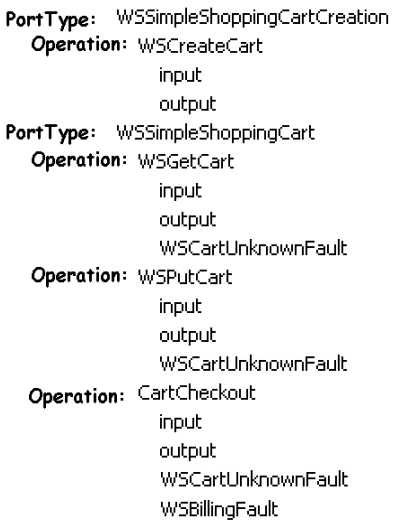

Figure 8: Web service defined by WSSimpleShoppingCart.wsdl

The operations are illustrated in detail below.

5.1.2. WSSimpleShoppingService Messages

The "WSSimpleShoppingService" Web service location is described by the URL in the EPR which references it. The cart is identified by a parameter returned from a WSCreateCart operation which is included in any message subsequently sent to the service. For example, the SOAP envelope for the WSCreateCart (and all other operations) is:

```
<SOAP-ENV:Header>
    . . .
    <wsa:To SOAP-ENV:mustUnderstand="1">
        http://www.example.com/WSSimpleShoppingService
    </wsa:To>
</SOAP-ENV:Header>
```

While the SOAP body of the request contains:

```
<SOAP-ENV:Body>
  <ws-ssc:WSCartCreateRequest>
   <ws-ssc:ProductCode>Cat-A2004-87968556</ws-ssc:ProductCode>
   <ws-ssc:Quantity>1</ws-ssc:Quantity>
  </ws-ssc:WSCartCreateRequest>
</SOAP-ENV:Body>
```

The SOAP body of the reply message is:

```
<SOAP-ENV:Body>
  <ws-ssc:WSCartCreateResponse
   <ws-ssc:Cart>S1</ws-ssc:Cart>
   <ws-ssc:ServiceAddress>
     <wsa:Address>http://example.com/ShoppingService</wsa:Address>
   </ws-ssc:ServiceAddress>
  </ws-ssc:WSCartCreateResponse>
</SOAP-ENV:Body>
```

The client takes the ServiceAddress and associates it with its representation of the service. The cart identifier "S1" must be subsequently copied into any call to the service as a parameter. This requirement to a requester to copy the identifier must be provided in documentation which accompanies the service definition. This procedure creates messages such as the following:

```
<ws-ssc:WSGetCart>
<ws-ssc:Cart>S1</ws-ssc:Cart>
</ws-ssc:WSGetCart>
```

The service responds with an XML representation of the shopping cart in the WSGetCartResponse message:

```
<SOAP-ENV:Body>
<ws-ssc:WSGetCartResponse>
 <ws-ssc:WSSimpleShoppingCart>
  <ssc:Item>
    <ssc:ProductCode>Cat-A2004-87968556</ssc:ProductCode>
    <ssc:Description>Garden String - 150m</ssc:Description>
    <ssc:Quantity>1</sc:Quantity>
    <ssc:ProductPrice>1.59</ssc:ProductPrice>
  </ssc:Item>
</ws-ssc:WSSimpleShoppingCart>
</ws-ssc:WSGetCartResponse>
</SOAP-ENV:Body>
```

5.2. Using WSRF Web Services

A service which exploits WSRF must define a Resource Properties document and may then use a combination of standard and service-specific operation definitions to define the messages which interact with the document and the resource which it describes.

5.2.1. The Resource Properties Document

There are two significant differences in the definition of a service which exploits WSRF, compared to the non-WSRF Web service illustrated in section 5.1.

- Firstly, WSRF associates an explicit XML description of the SimpleShoppingCart with the WSDL portType which defines the Web service operations on the cart. This description is known as the Resource Properties document. Figure 9 shows the relationship of the portType with an XML element which defines the root of the document, and the corresponding XML syntax of the association is illustrated in Figure 10.

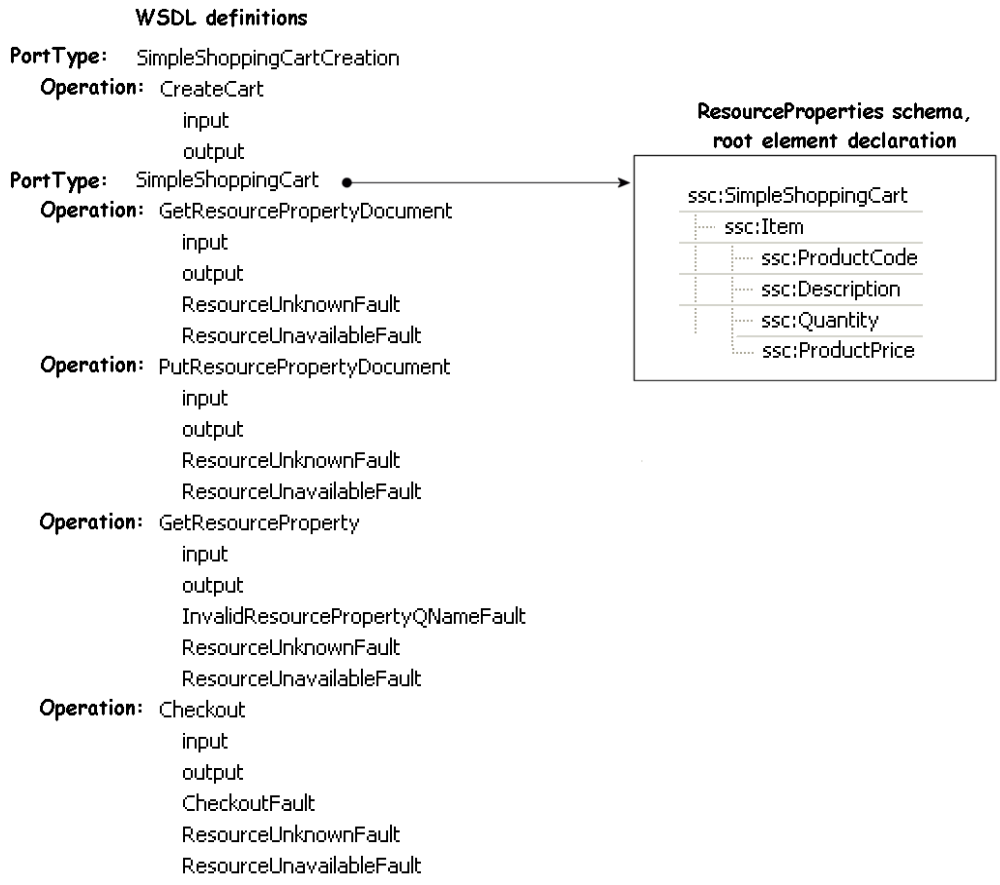

Figure 9: Web service defined by SimpleShoppingCart.wsdl

```
<wsdl:definitions>

 <wsdl:import
  namespace="http://docs.oasis-open.org/wsrf/rpw-2"
  location=" http://docs.oasis-open.org/wsrf/rpw-2.wsdl"/>

. . .

 <wsdl:types>
  <xsd:schema . . .>
   <xsd:import . . .
     schemaLocation="./SimpleShoppingCart.xsd" />
  </xsd:schema>
 </wsdl:types>

. . .

 <wsdl:portType name="SimpleShoppingCart"
           wsrf-rp:ResourceProperties="ssc:SimpleShoppingCart">
  <wsdl:operation name="GetResourcePropertyDocument">
   <wsdl:input message="wsrf-rpw:GetResourcePropertyDocumentRequest" . . />
   <wsdl:output message="wsrf-rpw:GetResourcePropertyDocumentResponse" . ./>

. . .

  </wsdl:operation>
. . .

 </wsdl:portType>

<wsdl:definitions>
```

Figure 10: Resource Properties document schema identification

- The `<wsdl:import>` element imports the WSRF specification definitions for the standard WSRF message definitions. The `<xsd:import>` element imports the SimpleShoppingCart schema which contains the SimpleShoppingCart element definition.
- The second difference is that the definition of simple operations such as a 'get' and 'put' can be completely standard: in Figure 10, the message definitions referenced by the 'wsrf-rpw' prefix are provided by the WSRF specifications. The example below illustrates the create operation which is specific to the service, and the standard GetResourcePropertyDocument operation.

The cart is created by the CreateCart request, just as in section 5.1. However, in the case of WSRF, the content of the message is simply an EPR and the identifier for the cart returned in the response message is part of the EPR:

```
<ssc:CartCreateResponse>
  <wsa:Address>http://www.example.com/SimpleShoppingService</wsa:Address>
  <wsa:ReferenceParameters>
    <rpimpl:CartId>S1</rpimpl:CartId>
  <wsa:ReferenceParameters>
</ssc:CartCreateResponse>
```

The contents of the EPR are generated by the service provider and are not intended to be understood by the client. The identifier for the cart in this example is `S1`.

The client takes the EPR from the response message and associates it with the client's programmatic representation of the cart (for example, a program object, reference or variable) so that all calls which use that representation are directed to the cart.

The GetResourcePropertyDocument operation can then be used to get the document. The SOAP envelope for the GetResourcePropertyDocument (and all other operations on the SimpleShoppingCart) contains:

```
<SOAP-ENV:Header>
    . . .
    <wsa:To SOAP-ENV:mustUnderstand="1">
        http://www.example.com/SimpleShoppingService
    </wsa:To>
    <rpimpl:CartId wsa:IsReferenceParameter="true">S1</rpimpl:CartId>
</SOAP-ENV:Header>
```

While the SOAP body of the request contains:

```
<SOAP-ENV:Body>
  <wsrf-rp:GetResourcePropertyDocument/>
</SOAP-ENV:Body>
```

The service returns a similar response to the equivalent request from section 5.1:

```
<SOAP-ENV:Body>
   <wsrf-rp:GetResourcePropertyDocumentResponse>
    <ssc:SimpleShoppingCart>
      <ssc:Item>
       <ssc:ProductCode>Cat-A2004-87968556</ssc:ProductCode>
       <ssc:Description>Garden String - 150m</ssc:Description>
       <ssc:Quantity>1</ssc:Quantity>
       <ssc:ProductPrice>1.59</ssc:ProductPrice>
      </ssc:Item>
    </ssc:SimpleShoppingCart>
   </wsrf-rp:GetResourcePropertyDocumentResponse>
</SOAP-ENV:Body>
```

Note that, in Figure 10, the definition of the GetResourcePropertyDocument operation and its message types are defined by the WS-ResourceProperties specification [WS-ResourceProperties] and its WSDL syntax can be based on the WSDL described in the specification.

6. Updating the Simple Shopping Cart

The SimpleShoppingCart document can be updated locally by the client and then sent back to the shopping service for validation and processing. An update might consist of a change in the quantity of a product in the cart.

6.1. Using non-WSRF Web Services

A non-WSRF Web service input message to replace the shopping cart document contains two items of information:

- An element containing the id of the cart.
- An element containing a modified version of the document.

Faults must be provided to diagnose errors in the replacement operation, such as errors in the document structure or content.

6.2. WSRF PutResourcePropertyDocument

WSRF provides this operation in a standard form which is described by the WS-ResourceProperties specification [WS-ResourceProperties]. It sends a new version of the Resource Properties document to the WS-Resource. For example, the following request contains a version of the SimpleShoppingCart document which has an increased quantity of the “Garden String” product compared to the one in section 5.

```
<SOAP-ENV:Body>
   <wsrf-rp:PutResourcePropertyDocumentRequest>
    <ssc:SimpleShoppingCart>
     <ssc:Item>
      <ssc:ProductCode>Cat-A2004-87968556</sc:ProductCode>
      <ssc:Description>Garden String - 150m</sc:Description>
      <ssc:Quantity>2</sc:Quantity>
       <ssc:ProductPrice>1.59</ssc:ProductPrice>
     </ssc:Item>
    </ssc:SimpleShoppingCart>
   </wsrf-rp:PutResourcePropertyDocumentRequest>
</SOAP-ENV:Body>
```

The WS-Resource compares the updated information against its own copy of the Resource Properties document and replies with either a simple confirmation that the

request was successful, or a copy of the document as it is represented by the WS-Resource following the operation.

The simple confirmation indicates that the document has the information requested by the client:

```
<SOAP-ENV:Body>
   <wsrf-rp:PutResourcePropertyDocumentResponse/>
</SOAP-ENV:Body>
```

If the service wishes to update the document in some other way than the update requested in the input message, it must respond with a copy of the updated document. For example, the increase in the quantity of an item might result in a free additional product being included in the cart: the whole SimpleShoppingCart document is sent back to the requester in the `<wsrf-rp:PutResourcePropertyDocumentResponse>` with both updates included.

The operation can also respond with the faults:

- ResourceUnknownFault
- ResourceUnavailableFault
- UnableToPutResourcePropertyDocument

These are described in sections 7.2 and 7.3.

7. Faults in WSRF

Each operation in the WSRF SimpleShoppingService or its non-WSRF Web services equivalent defines faults which may be sent by the service instead of the output message. This section summarises the facilities in WSRF for defining faults as specified in the Web Services BaseFaults specification [WS-BaseFaults].

7.1. The BaseFaultType

WSRF provides a recommended basic fault message element type from which all service-specific faults can be derived. The advantage of a common basic type is that all faults can, by default, contain common information. This is useful in complex systems where faults may be systematically logged, or forwarded through several layers of software and intermediate services before being analysed.

The common information includes:

- A mandatory timestamp.
- An element which can be used to indicate the originator of the fault.
- Other elements which can describe and classify the fault.

An example of the BaseFault element is shown below.

```
<wsrf-bf:BaseFault>
    <wsrf-bf:Timestamp>2005-05-31T12:00:00.000Z</wsrf-bf:Timestamp>
    <wsrf-bf:Originator>
     <wsa:Address>http://www.example.com/SimpleShoppingService</wsa:Address>
     <wsa:ReferenceParameters>
       <rpimpl:CartId>S1</rpimpl:CartId>
     </wsa:ReferenceParameters>
  </wsrf-bf:Originator>
   <wsrf-bf:Description>Descriptive Text</wsrf-bf:Description>
</wsrf-bf:BaseFault>
```

WS-Resource service definitions can use the XML `<extension>` mechanism to derive a fault with an appropriate name, and may also add additional, operation-specific information. For example, the following XML creates a new CheckoutFaultType for use in response to a CheckoutRequest:

```
<xsd:complexType name="CheckoutFaultType">
  <xsd:complexContent>
   <xsd:extension base="wsrf-bf:BaseFaultType">
    <xsd:sequence>
      <xsd:element name="CheckoutFaultDetails" type="xsd:string"/>
    </xsd:sequence>
   </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
```

Each fault of this type contains the information from the BaseFault and the CheckoutFaultDetails element defined by the extension:

```
<ssc:CheckoutFault>
  <wsrf-bf:Timestamp>2005-05-31T12:00:00.000Z</wsrf-bf:Timestamp>
  <wsrf-bf:Originator>
     <wsa:Address>http://www.example.com/SimpleShoppingService</wsa:Address>
     <wsa:ReferenceParameters>
      <rpimpl:CartId>S1</rpimpl:CartId>
     </wsa:ReferenceParameters>
  </wsrf-bf:Originator>
  <wsrf-bf:Description>Faulty Checkout Information</wsrf-bf:Description>
 <ssc:CheckoutFaultDetails>Credit Limit Exceeded!</ssc:CheckoutFaultDetails>
</ssc:CheckoutFault>
```

7.2. The Universal Faults

Two standard faults are defined for use with every WSRF operation. These are:

ResourceUnknownFault

- Used to indicate that the WS-Resource is not known by the service which receives the message.

ResourceUnavailableFault

- Used to indicate that the Web service is active, but unable to provide access to the resource.

When transmitted as a SOAP message, a typical example of the ResourceUnknownFault looks like this:

```
<SOAP-ENV:Body>
 <SOAP-ENV:Fault>
  <faultcode>SOAP-ENV:Client</faultcode>
  <faultstring>No such resource exists</faultstring>
  <faultactor>http://www.example.com/SimpleShoppingService</faultactor>
  <detail>
   <wsrf-r:ResourceUnknownFault>
    wsrf-bf:Timestamp>2005-05-31T12:00:00.000Z</wsrf-bf:Timestamp>
    <wsrf-bf:Originator>
     <wsa:Address>http://www.example.com/SimpleShoppingService</wsa:Address>
     <wsa:ReferenceParameters>
       <rpimpl:CartId>S0</rpimpl:CartId>
     </wsa:ReferenceParameters>
    </wsrf-bf:Originator>
    <wsrf-bf:Description>Resource unknown</wsrf-bf:Description>
   </wsrf-r:ResourceUnknownFault>
  </detail>
 </SOAP-ENV:Fault>
</SOAP-ENV:Body>
```

7.3. Operation-specific Faults

7.3.1. Faults Defined by WSRF Specifications

WSRF operations declare fault messages which must be copied into the SimpleShoppingService definition along with the input and output message declarations. For example, the PutResourcePropertyDocument operation defines the fault 'UnableToPutResourcePropertyDocument' which occurs if the WS-Resource was unable to complete the processing of the PutResourcePropertyDocument for some reason.

7.3.2. Faults Defined by a WSRF Service Application

Each non-standard operation of a service which exploits WSRF may need to define its own faults. The Checkout operation of the SimpleShoppingCart defines its own fault, decribed in section 7.1 to allow details of any problem to be reported to the client. It is defined as an extension of the WSRF BaseFaultType so that it can be processed by a standard fault-reporting component.

8. Resource Properties and the Printer Service

The Resource Properties document of a WS-Resource may be defined as a sequence of named elements and an instance of the document is accessed through the WS-Resource. These individually named elements can be retrieved and updated by standard WSRF operations which are described below and specified in the WS-ResourceProperties specification [WS-ResourceProperties]

The Printer service illustrated in Figure 11 is a typical network printer. It accepts messages which control and exploit its facilities. Its state consists of a number of control and status variables, such as the length of the printer job queue, and is described in XML. The Printer is described in two forms so that the facilities offered by WSRF are clear: one form is based on non-WSRF Web services, the second on WSRF.

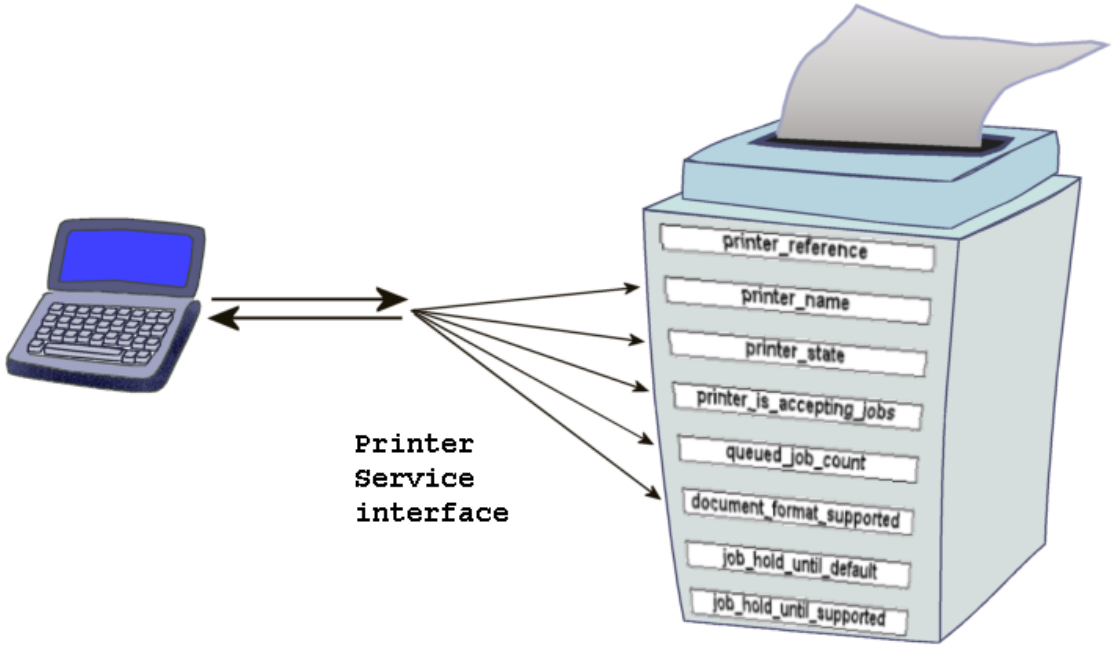

Figure 11: The Printer Service

The examples below illustrate query and manipulation of the status and control variables. Queries of these values might be executed by users of the printer to detect whether the printer is operational (`printer_is_accepting_jobs` is set to 'true'), and whether there is a queue of outstanding work (`queued_job_count` is non-zero). Set operations are likely to be used by administrators to control the status and capabilities of the printer, but note that the messages defined by WSRF are not concerned with the issue of distinguishing normal users from administrators, nor controlling access to the operations: those facilities can be provided using other Web services standards.

The printer service must also support operations to accept print jobs from users and answer queries about those jobs. The construction of the Printer and the resource properties describing the print jobs is developed in section 10.

8.1. The Printer Resource Properties

The state of a printer is described by a set of values such as the ones in Table 1 (below).

Element name	Element value
pr:Printer	
pr:printer_reference	
wsa:Address	http://www.example.com/Printer
wsa:ReferenceParameters	<rpimpl:PrinterId>Printer01 </rpimpl:PrinterId>
pr:printer_name	Printer01
pr:printer_state	Idle
pr:printer_is_accepting_jobs	true
pr:queued_job_count	0
pr:document_format_supported	
pr:mimeMediaType	text/plain
pr:job_hold_until_default	No-Hold
pr:job_hold_until_supported	No-Hold
pr:job_hold_until_supported	Overnight

Table 1: Printer resource property values

Indentation of the element names in the table shows how the level of their containment within the document. pr:Printer is the root element. Each of its direct child elements (pr:printer_reference, pr:printer_name and so on) have global declarations in the 'pr' namespace, and can be referenced from other XML documents. They are known as the *resource properties* of the WS-Resource. Messages defined by the WS-Resource Properties specification [WS-ResourceProperties] use these global names to access the resource property values.

The properties are defined by the schema for the pr:PrinterRP document. The number of times each element may occur in the description of a printer is summarized in the table of multiplicities below.

PropertyName	Multiplicity
pr:printer_reference	1
pr:printer_name	1
pr:printer_state	1
pr:printer_is_accepting_jobs	1
pr:queued_job_count	1
pr:document_format_supported	1
pr:job_hold_until_default	0..1
pr:job_hold_until_supported	0..*
wsrf-rp:QueryExpressionDialect	0..*
pr:job_properties	0..*

Table 2: Printer property multiplicities

The meanings of the printer resource properties are:

- printer_reference
  An EndpointReference used to identify the printer. This EPR is copied into jobs created by the printer and can be used subsequently to find out about the state of the printer.
- printer_name
  A human-readable identification for the printer
- printer_state
  Describes whether the printer is healthy.
- printer_is_accepting_jobs
  'true' if the printer is open for business, otherwise 'false'.
- queued_job_count
  The number of jobs queued for the printer.
- document_format_supported
  Types of input file format acceptable to the printer.
- job_hold_until_default
  If the printer supports time slots in which jobs can be printed, this element defines the default time slot. Its value must be the same as one of the 'supported' values.
- job_hold_until_supported
  Defines elements which describe a number of named time slots. If there are slots defined, a default slot must be present in the 'job_hold_until_default' element.
- wsrf-rp:QueryExpressionDialect
  This can be used to specify that the printer supports a query language different

from the default defined by QueryResourceProperties operation defined in [WS-ResourceProperties] It is not used in this example.

- job_properties
  Each of these element values describes a job created by the printer.

8.2. GetResourceProperty

The value or values associated with a named resource property can be retrieved from a WS-Resource using the GetResourceProperty operation. For example,

```
 <SOAP-ENV:Body>
   <wsrf-rp:GetResourceProperty>pr:queued_job_count</wsrf-
rp:GetResourceProperty>
 </SOAP-ENV:Body>
```

returns the response:

```
<SOAP-ENV:Body>
  <wsrf-rp:GetResourcePropertyResponse>
    <pr:queued_job_count>0</pr:queued_job_count>
  </wsrf-rp:GetResourcePropertyResponse>
</SOAP-ENV:Body>
```

If the Resource Properties document contains multiple value elements for the property, all the values are returned:

```
 <SOAP-ENV:Body>
   <wsrf-rp:GetResourceProperty>pr:job_hold_until_supported</wsrf-
rp:GetResourceProperty>
 </SOAP-ENV:Body>
```

returns

```
<SOAP-ENV:Body>
  <wsrf-rp:GetResourcePropertyResponse>
    <pr:job_hold_until_supported>No hold</pr:job_hold_until_supported>
    <pr:job_hold_until_supported>Overnight</pr:job_hold_until_supported>
  </wsrf-rp:GetResourcePropertyResponse>
</SOAP-ENV:Body>
```

If the Resource Properties document contains no values element for the property, an empty response is returned. For example, since there are no `pr:job_properties` in the document, a GetResourceProperty request returns:

```
<SOAP-ENV:Body>
  <wsrf-rp:GetResourcePropertyResponse>
  </wsrf-rp:GetResourcePropertyResponse>
</SOAP-ENV:Body>
```

8.3. Resource Property Access Faults

Sections 8.5 and onwards describe other operations to retrieve and modify resource properties. The faults associated with many of these operations follow a common pattern, using the following fault definitions:

- InvalidResourcePropertyQNameFault
  This fault is returned if a property named in the request message does not correspond to a resource property element.

For operations which attempt to modify resource properties, the following may apply.

- UnableToModifyResourcePropertyFault
  This fault is returned if a request is made to modify a resource property which is not modifiable by the user. The restriction on modifiability may be part of

the implementation logic, or it could be described by some means outside of the scope of WSRF.

- InvalidModificationFault
  This fault is returned if a request is made to modify properties which would cause the Resource Properties document to fail validation.

In faults which report a failure to modify the resource properties, the fault message can be used to describe whether the Resource Properties document was restored to its original state. If several modifications were being attempted, the fault message can report which one failed. An example of this is described in section 8.10.1.

Operations may also have a catch-all fault for other errors, and the QueryResourceProperties and SetResourceProperties operations also have specific faults associated with their syntax. These are described in sections 8.6.1 and 8.10.1.

8.4. Equivalent Non-WSRF services

The XML document which describes the resource properties for the printer could also be used in a non-WSRF Web services implementation of the printer service to construct explicit message types based on the same information. The advantage in using WSRF over this method can be seen by comparing the message definitions in the WSDL and schema files which accompany the Primer as described in section 14. For example, the GetResourceProperty operation based on [WS-ResourceProperties] has the same function as the WSGet_Printer_Attribute operation defined in the non-WSRF Web service described by WSPrinter.wsdl and WSPrinter.xsd.

In the following request, the printer is identified explicitly as an element in the request message, and the syntax of the message has a definition specific to the printer service.

```
<ws-pr:Get_Printer_Attribute_Request>
    <pr:printer_reference>
     <wsa:Address>http://www.example.com/Printer</wsa:Address>
     <wsa:ReferenceParameters>
      <rpimpl:PrinterId>Printer01</rpimpl:PrinterId>
     </wsa:ReferenceParameters>
    </pr:printer_reference>
    <ws-pr:Get_Printer_Attribute>printer_name</ws-pr:Get_Printer_Attribute>
</ws-pr:Get_Printer_Attribute_Request>
```

WSRF avoids the need for the service-specific definition of these messages and their faults, and the associated implementation.

8.5. GetMultipleResourceProperties

This operation retrieves multiple property values. The request:

```
<SOAP-ENV:Body>
 <wsrf-rp:GetMultipleResourceProperties>

<wsrf-rp:ResourceProperty>pr:printer_state</wsrf-rp:ResourceProperty>
<wsrf-rp:ResourceProperty>pr:queued_job_count</wsrf-rp:ResourceProperty>

</wsrf-rp:GetMultipleResourceProperties >

</SOAP-ENV:Body>
```

produces the response

```
<SOAP-ENV:Body>
   <wsrf-rp:GetMultipleResourcePropertiesResponse>
         <pr:printer_state>Idle</pr:printer_state>
         <pr:queued_job_count>0</pr:queued_job_count>
   </wsrf-rp:GetMultipleResourcePropertiesResponse>
</SOAP-ENV:Body>
```

8.6. QueryResourceProperties

The QueryResourceProperties operation allows the execution of a query expression on the Resource Properties document. The following example uses the XPath version 1.0 query language which is always supported by the operation.

```
<SOAP-ENV:Body>
   <wsrf-rp:QueryResourceProperties>
    <wsrf-rp:QueryExpression
        Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
           contains(/*/*/*[namespace-uri()=
           'http://docs.oasis-open.org/wsrf/Printer.xsd' and
           local-name()='mimeMediaType'],"text/plain")
    </wsrf-rp:QueryExpression>
   </wsrf-rp:QueryResourceProperties>
</SOAP-ENV:Body>
```

The operation returns a response similar to the following, indicating that the XPath expression evaluated to 'true' since the printer does support the mimeMediaType of 'text/plain' for document input:

```
<SOAP-ENV:Body>
   <wsrf-rp:QueryResourcePropertiesResponse>true</wsrf-
rp:QueryResourcePropertiesResponse>
</SOAP-ENV:Body>
```

8.6.1. QueryResourceProperties Faults

In addition to the standard faults described in section 7.2, QueryResourceProperties may respond with the following faults:

- UnknownQueryExpressionDialectFault
  The given QueryExpression has a dialect that is unknown to the Web service.
- InvalidQueryExpressionFault
  The given QueryExpression is not valid within the QueryExpression language identified by the dialect attribute.
- QueryEvaluationErrorFault
  The Query Expression failed during evaluation.

8.7. UpdateResourceProperties

This operation can be used to change the value element(s) of a property. The following administration request prevents the printer from accepting new jobs.

```
<SOAP-ENV:Body>
<wsrf-rp:UpdateResourceProperties>
  <wsrf-rp:pdate>
   <pr:printer_is_accepting_jobs>false</pr:printer_is_accepting_jobs>
  </wsrf-rp:Update>
 </wsrf-rp:UpdateResourceProperties>
</SOAP-ENV:Body>
```

The successful response is:

```
<SOAP-ENV:Body>
  <wsrf-rp:UpdateResourcePropertiesResponse/>
</SOAP-ENV:Body>
```

If there are multiple value elements for the property, all values are replaced by the values contained in the update message. For example, the following update operation replaces the two values for the `job_hold_until_supported` property shown in Indenta with a single value.

```
<SOAP-ENV:Body>
 <wsrf-rp:UpdateResourceProperties>
  <wsrf-rp:Update>
   <pr:job_hold_until_supported>No hold</pr:job_hold_until_supported>
  </wsrf-rp:Update>
 </wsrf-rp:UpdateResourceProperties>
</SOAP-ENV:Body>
```

8.8. InsertResourceProperties

An Insert request adds a new value element, or several elements for a resource property. The following request adds new value elements for the job_hold_until_supported property:

```
<SOAP-ENV:Body>
 <wsrf-rp:InsertResourceProperties>
   <wsrf-rp:Insert>
    <pr:job_hold_until_supported>Morning</pr:job_hold_until_supported>
    <pr:job_hold_until_supported>Afternoon</pr:job_hold_until_supported>
   </wsrf-rp:Insert>
 </wsrf-rp:InsertResourceProperties>
</SOAP-ENV:Body>
```

8.9. DeleteResourceProperties

A Delete request removes all value elements for the resource property named in the request. For example, the following request attempts to remove all values of the job_hold_until_supported property:

```
<SOAP-ENV:Body>
   <wsrf-rp:DeleteResourceProperties>
       <wsrf-rp:Delete ResourceProperty="pr:job_hold_until_supported"/>
   </wsrf-rp:DeleteResourceProperties>
</SOAP-ENV:Body>
```

However, this request will respond with DeleteResourcePropertiesRequestFailedFault because of the requirement that the job_hold_until_default should not exist without a corresponding job_hold_until_supported property. The next section describes the SetResourceProperties operation which provides a way to modify multiple properties.

8.10. SetResourceProperties

The SetResourceProperties operation allows a sequence of changes to a Resource Properties document to be batched together instead of using discrete Insert, Update and Delete operations.

For example, to achieve a change in the definition of the time slots in which a document is printed by the printer, it is necessary to change two related resource properties.

Suppose the required change is from:

pr:job_hold_until_default	No-Hold
pr:job_hold_until_supported	No-Hold
pr:job_hold_until_supported	Overnight

to:

pr:job_hold_until_default	Daytime
pr:job_hold_until_supported	Daytime
pr:job_hold_until_supported	Overnight

The following SetResourceProperties request will make the change:

```
<wsrf-rp:SetResourceProperties>

 <wsrf-rp:Update>
  <pr:job_hold_until_default>Daytime</pr:job_hold_until_default>
 </wsrf-rp:Update>

 <wsrf-rp:Update>
   <pr:job_hold_until_supported>Overnight</pr:job_hold_until_supported>
   <pr:job_hold_until_supported>Daytime</pr:job_hold_until_supported>
 </wsrf-rp:Update>

</wsrf-rp:SetResourceProperties>
```

8.10.1. SetResourceProperties Faults

WSRF defines the way in which the success, failure, or partial success of the sequence can be reported. An implementation of a WS-Resource description may be able to undo the batch of changes if one of the components fails. For example, the following request is invalid because it attempts to insert a value element for the printer_is_accepting_jobs property and the printer cannot have two values for this property:

```
<wsrf-rp:SetResourceProperties>

 <wsrf-rp:Update>
  <pr:job_hold_until_default>Daytime</pr:job_hold_until_default>
 </wsrf-rp:Update>

 <wsrf-rp:Update>
   <pr:job_hold_until_supported>Overnight</pr:job_hold_until_supported>
   <pr:job_hold_until_supported>Daytime</pr:job_hold_until_supported>
 </wsrf-rp:Update>

 <wsrf-rp:Insert>
   <pr:printer_is_accepting_jobs>false</pr:printer_is_accepting_jobs>
 </wsrf-rp:Insert>

</wsrf-rp:SetResourceProperties>
```

The fault details might be:

```
<wsrf-rp:InvalidModificationFault>
 <wsrf-bf:Timestamp>2005-12-31T12:00:00</wsrf-bf:Timestamp>
 <wsrf-rp:ResourcePropertyChangeFailure Restored="true">

 <wsrf-rp:CurrentValue>
    <pr:printer_is_accepting_jobs>true</pr:printer_is_accepting_jobs>
 </wsrf-rp:CurrentValue>
```

```
 <wsrf-rp:RequestedValue>
    <pr:printer_is_accepting_jobs>false</pr:printer_is_accepting_jobs>
 </wsrf-rp:RequestedValue>

</wsrf-rp:ResourcePropertyChangeFailure>
</wsrf-rp:InvalidModificationFault>
```

The CurrentValue and RequestedValue elements identify the faulty modification and the 'Restored="true"' attribute reports that the values of job_hold_until_default and job_hold_until_supported are restored to the state that existed before the request was received.

9. The Shopping Cart and WS-Resource Lifetime

The SimpleShoppingCart used in section 5 was updated with an operation which replaced the whole cart as a single unit. This section uses a more complex document structure to enable the retrieval and update of one of the items in the cart. This is done by representing each item as WS-Resource.

This section also illustrates a facility provided by WSRF for destroying the Cart if, within a reasonable time, there is no checkout instruction from the requester. This facility is representative of the general requirement in many systems where resources are allocated on an 'as needed' basis, and must be cleaned up when the need is no longer apparent.

9.1. Structure of the Shopping Cart

The ShoppingCart schema has the following structure.

Element Name		Multiplicity
sc:TotalPrice		1
sc:Item		0..*
	sc:ItemEPR	1
	sc:ProductCode	1
	sc:Description	1
	sc:Quantity	1
	sc:ProductPrice	1
wsrf-rl:CurrentTime		1
wsrf-rl:TerminationTime		1

Table 3: Shopping Cart schema

Compared to the SimpleShoppingCart in Figure 9, the ShoppingCart adds the Properties:

- sc:TotalPrice which is the total cost of the cart
- wsrf-rl:CurrentTime and wsrf-rl:TerminationTime which are used to control the lifetime of the cart.

Also, for each Item property in the cart, an ItemEPR property is added as a reference to the Item.

The following example shows an instance of the ShoppingCart Resource Properties document.

```
<?xml version="1.0" encoding="UTF-8"?>
<sc:ShoppingCart  . . . >
  <sc:TotalPrice>18.17</sc:TotalPrice>
  <sc:Item>
      <sc:ItemEPR>
        <wsa:Address>http://www.example.com/Shoppingcart/Item</wsa:Address>
        <wsa:ReferenceParameters>
         <rpimpl:CartId>Cart2005</rpimpl:CartId>
         <rpimpl:ItemId>Item1</rpimpl:ItemId>
        </wsa:ReferenceParameters>
      </sc:ItemEPR>
      <sc:ProductCode>Cat-A2004-87968556</sc:ProductCode>
      <sc:Description>Garden String - 150m</sc:Description>
      <sc:Quantity>2</sc:Quantity>
      <sc:ProductPrice>1.59</sc:ProductPrice>
  </sc:Item>
  <sc:Item>
      <sc:ItemEPR>
        <wsa:Address>http://www.example.com/ShoppingCart/Item</wsa:Address>
        <wsa:ReferenceParameters>
         <rpimpl:CartId>Cart2005</rpimpl:CartId>
         <rpimpl:ItemId>Item2</rpimpl:ItemId>
        <wsa:ReferenceParameters>
      </sc:ItemEPR>
      <sc:ProductCode>Cat-A2004-00109836</sc:ProductCode>
      <sc:Description>Fishing Gnome - 20cm</sc:Description>
      <sc:Quantity>1</sc:Quantity>
      <sc:ProductPrice>14.99</sc:ProductPrice>
  </sc:Item>
  <wsrf-rl:CurrentTime>2005-12-31T12:00:00</wsrf-rl:CurrentTime>
  <wsrf-rl:TerminationTime>2005-12-31T23:59:00</wsrf-rl:TerminationTime>
</sc:ShoppingCart>
```

The operations available on the ShoppingCart are:

- GetResourcePropertyDocument, GetResourceProperty – as described in the preceding sections. These operations can be used to retrieve the complete ShoppingCart document, or a single property, such as the TotalPrice.
- AddItem – used to add new items to the cart.
- Checkout
- Destroy – used to destroy the cart without checking it out.
- SetTerminationTimeRequest – used to extend the lifetime of a shopping cart.

The AddItem request returns an EndpointReference (ItemEPR), which can then be accessed as a WS-Resource in its own right using the element `sc:Item` Resource Properties document. The schema for the `sc:Item` element is shown in ent Name.

The operations available on the Item are:

- GetResourcePropertyDocument – retrieves the details of the Item.
- UpdateResourceProperties – enables update of the Quantity property of the Item.
- Destroy – removes the item from the shopping cart.

Note that the 'Item' and 'ShoppingCart' WS-Resources are different views of the same shopping cart information: if the quantity of a product is increased via the Item WS-Resource, the ShoppingCart document is also modified, and the TotalPrice property of the ShoppingCart document is increased accordingly by the service implementation and can be retrieved with the GetResourceProperty request to the ShoppingCart WS-Resource.

9.2. Adding and Updating Items

The ShoppingCart WS-Resource provides an AddItem operation which takes a product code and quantity and returns an EndpointReference (EPR) to the new Item. The EPR can be used to construct requests for the new Item, including the ability to modify the quantity of the product in the Item.

9.3. The Destroy Operation

A useful operation is the ability to explicitly destroy a resource that is no longer needed. In the case of the shopping cart, it is useful to remove unwanted items but, in general, a WS-Resource may represent physical resources which should be made available for reuse. WSRF provides a standard Destroy operation which can be used to request that a WS-Resource no longer respond to requests. The form of the request is:

```
<SOAP-ENV:Body>
  <wsrf-rl:Destroy/>
</SOAP-ENV:Body>
```

9.4. ShoppingCart Lifecycle and Lifetime

Many resources have a naturally limited lifetime. In the case of the ShoppingCart, the lifetime is the time that is available for a shopper to decide to checkout the cart (somewhere between a few hours and a few days) before it is deleted on the presumption that the shopper decided not to buy the products. This allows the implementation of the cart to clean-up and recycle storage and release reserved inventory.

WSRF provides the resource property `wsrf-rl:TerminationTime` as a standard mechanism to record the lifetime of a WS-Resource. The `wsrf-rl:CurrentTime` property allows account to be taken of any difference in the local time. When the TerminationTime is reached, the shopping cart may be removed.

The SetTerminationTimeRequest operation can be used to request a change in the TerminationTime in order to delay or advance the destruction of the shopping cart. This feature is known as "scheduled destruction".

10.WS-Resources as Views of Underlying Resources

Section 3 described the way in which WS-Resources present, to a Web services requester, a view of some underlying information resource. The shopping service of section 9 presents multiple views: a consolidated view is described by the ShoppingCart WS-Resource while each instance of the Item WS-Resource describes a restricted view, namely one item within the cart. The same idea of a compound information resource with multiple views can be used to describe the Printer of section 8 and its Jobs. The concept that the 'Printer and Jobs' can be one coherent resource has been adopted in this Primer because it follows the model described by the Internet

Printing Protocol [RFC 2566]. It also illustrates that the relationship between a WS-Resource, its underlying information resources and any physical devices which it represents may be complex, as shown in Figure 12.

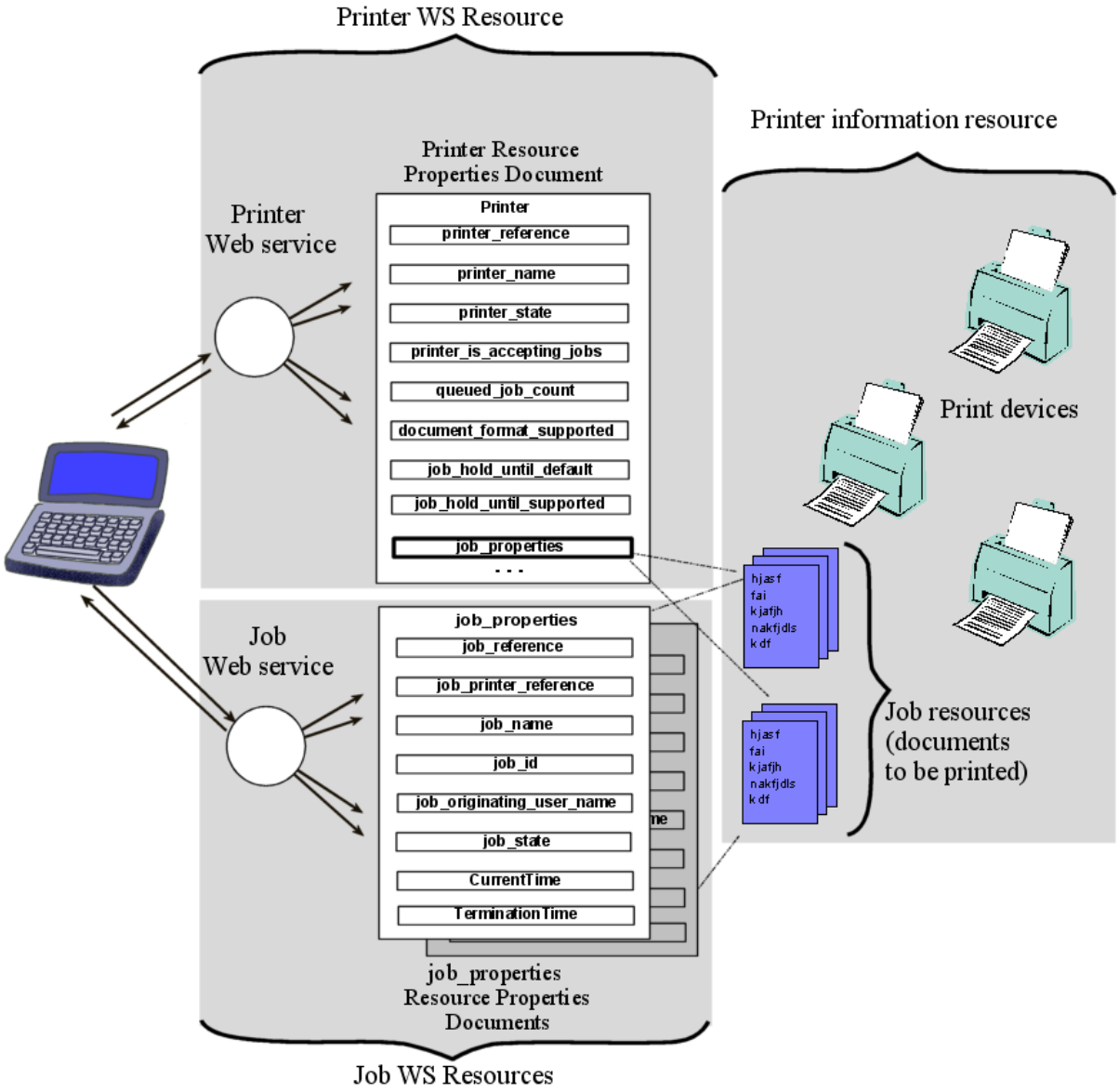

Figure 12: The printer resource and its WS-Resources

The Resource Properties document of the Printer WS-Resource contains information describing the state of:

- A logical printer which may represent several physical output devices.
- Multiple print jobs described by the job_properties (highlighted) resource property values.

The Printer Web service portType defines the operations to access resource properties described in section 8, but also defines PrintJob and CreateJob operations which both create a new Job WS-Resource: PrintJob allows a single document to be printed, CreateJob allows for multiple documents to be added to the job.

Each of the print Jobs is individually accessible via a Job WS-Resource and each expresses the job information via a job_properties XML document. Note that the TerminationTime property is included in the job_properties to describe the lifetime of a Job. When a Job has been printed, the Job WS-Resource persists for short time so that its status can be seen by its originating user.

Like the Shopping Cart, all of the views of the Printer and its Jobs are coherent. Section 11 describes a way of organizing collective information which is not necessarily a coherent view.

11.Directories and Other Groups of Services

The WSRF ServiceGroup [WS-ServiceGroup] specification provides a description of a general-purpose WS-Resource which aggregates information about multiple WS-Resources or Web services. The aggregated information can be used as a directory in which the descriptive abstracts of the individual WS-Resources and Web services can be queried to identify useful entries.

The example in this section is based on the Printer and its Jobs which were introduced in sections 8 and 10. The PrinterAndJobGroup contains abstracts of information about several Printer WS-Resource and the Jobs associated with the Printers. This allows a requester to query the PrinterAndJobGroup to find (for example) a Printer that is idle, or one that has a small printer queue, or to find out whether Jobs previously submitted have finished printing. .

11.1. PrinterAndJobGroup Resource Properties

Figure 13 summarizes the form of the resource properties of the directory.

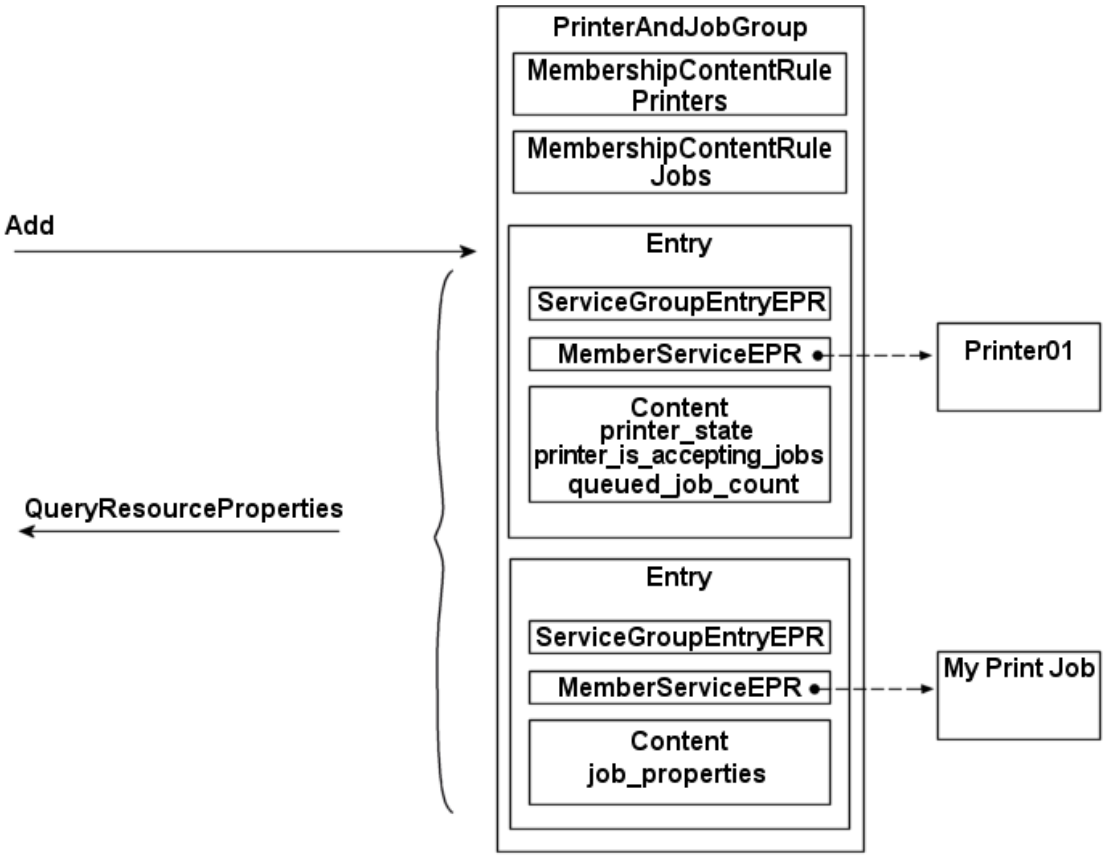

Figure 13: PrinterAndJobGroup resource properties and operations

The directory is described by a WSDL portType called PrinterAndJobGroup which contains the operations to add new entries and query existing entries using QueryResourceProperties. The Resource Properties document is shown in the figure; it contains the entries, along with rules which describe them.

11.1.1. MembershipContentRules

MembershipContentRules are special resource properties that are characteristic of a ServiceGroup. They describe the entries in the group and restrict membership of the group to Web services which have appropriate portTypes. In the case of this example there are two MembershipContentRules:

- Member services which support the pr:Printer portType must be described by an entry which contain three XML elements defined by the printer:

```
<wsrf-sg:MembershipContentRule
  MemberInterfaces="preprinted"
  ContentElements="pr:printer_state
                   pr:printer_is_accepting_jobs
                   pr:queued_job_count"
/>
```

  These XML elements are an abstract of the description of the corresponding Printer.

- Member services which support the pr:Job portType must be described by an entry which contains an XML element that is a copy of the complete Resource Properties document of the corresponding Job:

```
<wsrf-sg:MembershipContentRule
  MemberInterfaces="prw:Job"
  ContentElements="wsrf-sg:RPDoc"
/>
```

11.1.2. Entries

Each entry in the directory PrinterAndJobDirectory is either a Printer or a Job and each contains:

- An EndpointReference (EPR), called the MemberServiceEPR, which can be used to send requests to the Printer or Job.
- A ServiceGroupEntryEPR which identifies the Entry and can be used to modify or remove it.
- An abstract, called the `Content`, which is described by the appropriate MemberhipContentRule. In the case of a Printer entry, the Content is the three XML elements describing the printer. In the case of a Job entry, the Content is a job_properties element which is the root element of the Resource Properties document of a Job WS-Resource.

11.2. Searching the Directory Entries

The PrinterAndJobDirectory portType contains the QueryResourceProperties operation, which can be used to search for a MemberServiceEPR matching some interesting criteria. For example, the request:

```
<SOAP-ENV:Body>
   <wsrf-rp:QueryResourceProperties>
    <wsrf-rp:QueryExpression
     Dialect=”http://www.w3.org/TR/1999/REC-xpath-19991116”>

      //wsrf-sg:Entry
      /wsrf-sg:Content[child::pr:printer_state="Idle"]
      /../wsrf-sg:MemberServiceEPR

     </wsrf-rp:QueryExpression>
    </wsrf-rp:QueryResourceProperties>
</SOAP-ENV:Body>
```

returns the EPRs for all idle printers.

11.3. Adding New Printer Entries

Entries in the PrinterAndJobGroup are created using the Add operation. The direct effect is the creation of a new PrinterEntry or JobEntry WS-Resource which is based on the definition of ServiceGroupEntry [WS-ServiceGroup]. The existence of these WS-Resources is then reflected in the corresponding addition of a new Entry element to the resource properties of the PrinterAndJobGroup.

Note that the appearance of a new Entry in the PrinterAndJobGroup need not be completely synchronised with the creation of the PrinterEntry WS-Resource. This is different from, and less coherent than, the behaviour for the creation of new ShoppingCart Entries in section 9, or new jobs for a Printer WS-Resource in section 10. The specification of the ServiceGroup is more complex because it accommodates large-scale implementation where assembly of the PrinterAndJobGroup resource properties by the service implementation operates on copies of the information represented in the Entry WS-Resources.

The Add operation is used to carry the information for the resource properties of the new PrinterEntry WS-Resource, namely its MemberServiceEPR and descriptive Contents. For example, a WS-Resource corresponding to the Printer shown in Figure 13 can be added with the request:

```
<SOAP-ENV:Body>
 <wsrf-sg:Add>
  <wsrf-sg:MemberEPR> . (details omitted) . </wsrf-sg:MemberEPR>
  <wsrf-sg:Content>
     <pr:printer_state>Idle</pr:printer_state>
     <pr:printer_is_accepting_jobs>true</pr:printer_is_accepting_jobs>
     <pr:queued_job_count>0</pr:queued_job_count>
   </wsrf-sg:Content>
 </wsrf-sg:Add>
</SOAP-ENV:Body>
```

The resource properties of the PrinterEntry are similar, but not identical, to the corresponding Entry in the PrinterAndJobGroup:

```
<prjg:PrinterEntry>
 <wsrf-sg:ServiceGroupEPR>. (details omitted) . </wsrf-sg:ServiceGroupEPR>
 <wsrf-sg:MemberEPR> . (details omitted) .  </wsrf-sg:MemberEPR>
 <wsrf-sg:Content>
  <pr:printer_state>Idle</pr:printer_state>
  <pr:printer_is_accepting_jobs>true</pr:printer_is_accepting_jobs>
  <pr:queued_job_count>0</pr:queued_job_count>
 </wsrf-sg:Content>
 <wsrf-rl:CurrentTime>2005-12-31T12:00:00</wsrf-rl:CurrentTime>
 <wsrf-rl:TerminationTime xsi:nil="true"></wsrf-rl:TerminationTime>
</prjg:PrinterEntry>
```

In addition to the MemberEPR and Content elements, these resource properties contain:

- The ServiceGroupEPR property which is the EPR of the containing group.
- The resource properties needed to support the Scheduled Destruction behaviour described in section 9.4. These are a standard part of the ServiceGroupEntry. In the example above, the PrinterEntry WS-Resource has been created without a termination time: it can only be removed with an explicit Destroy request instigated by human activity when the printer service is removed from the network.

The response contains the ServiceGroupEntryEPR which identifies the new PrinterEntry WS-Resource and a ‘nil’ value for the scheduled destruction time.

```
<wsrf-sg:AddResponse>
   <wsrf-sg:ServiceGroupEntryReference> .(details omitted) .
   </wsrf-sg:ServiceGroupEntryReference>
   <wsrf-sg:TerminationTime xsi:nil="true"></wsrf-sg:TerminationTime>
   <wsrf-sg:CurrentTime>2005-12-31T12:00:00</wsrf-sg:CurrentTime>
</wsrf-sg:AddResponse>
```

In the example above, the creation and deletion of Printer Entries in the PrinterAndJobGroup could be the result of human activity reflecting the installation or removal of network printer services. As an alternative process for the deletion step, the TerminationTime property could be used by the implementation of the PrinterEntry WS-resource to describe the Scheduled Destruction of the PrinterEntry. Section 11.5 describes how the information in the Printer Entry could be automatically updated using Notification messages from the Printer WS-Resource. The receipt of these messages can be used by the PrinterEntry resource to continually extend the lifetime of the PrinterEntry. In this way, the removal of the printer service results in the automatic removal of the corresponding PrinterEntry in the directory of printers.

11.4. Adding New Job Entries

The Web service operation to create Job Entries in the PrinterAndJobGroup is similar to the one for Printers. There are two differences:

- The Add operation is an indirect result of a Print or CreateJob operation on a Printer WS-Resource. It is the implementation of the Printer WS-Resource which invokes Add operation on the PrinterAndJobGroup.
- The JobEntry is created with an initial lifetime reflecting the lifetime of the corresponding Job WS-Resource. The Add request is used to communicate the initial Termination time, which may be subsequently modified as a result of notification messages as described in section 11.5. The Job Entry is then destroyed at approximately the same time as the corresponding Job WS-Resource.

11.5. Updating the Entries: Notification Messages

To be useful, the entries in the PrinterAndJobGroup must contain up-to-date information about the state of the Printers and Jobs as they change. Interfaces which define the publication of such information to interested subscribers are defined by Web service specifications such as WS-BaseNotification [WS-BaseNotification]. In our example, the implementation of the Job WS-Resource must act as a publisher and allow the JobEntry WS-Resource to subscribe to changes in the values of the resource properties describing the job resource.

WSRF describes the format of several kinds of event which can be incorporated in notification messages defined by WS-BaseNotification. These are:

- Changes to resource property values defined in [WS-Resource Properties]
- WS-Resource termination, defined in [WS-Resource Lifetime]
- Addition and removal of ServiceGroupEntries, defined in [WS-ServiceGroup]

For example, the change to a job_state resource property could be communicated to the JobEntry WS-Resource in the following message, where WSRF defines the name of the Topic (pr:job_state, shown in bold type) on which the message is published, and

the element which describes the details of the change that has taken place (shown in bold).

```
<wsnt:Notify>
 <wsnt:NotificationMessage>
  <wsnt:SubscriptionReference>(details omitted)</wsnt:SubscriptionReference>
  <wsnt:Topic
    Dialect="http://docs.oasis-open.org/wsn/t-1/TopicExpression/Simple">
     pr:job_state
  </wsnt:Topic>
  <wsnt:ProducerReference>
   <wsa:Address>http://www.example.com/Printer/Job</wsa:Address>
   <wsa:ReferenceParameters>
     <rpimpl:PrinterId>Printer01</rpimpl:PrinterId>
     <rpimpl:JobId>Job523</rpimpl:JobId>
   </wsa:ReferenceParameters>
  </wsnt:ProducerReference>
  <wsnt:Message>

    <pr:Job_stateChangeNotification>
    <pr:printer_name>Printer01</pr:printer_name>
    <pr:job_name>My Print Job</pr:job_name>
     <wsrf-rp:ResourcePropertyValueChangeNotification>
       <wsrf-rp:OldValues>
         <pr:jobprinter_state>processing</pr:job_state>
       </wsrf-rp:OldValues>
       <wsrf-rp:NewValues>
         <pr:job_state>completed</pr:job_state>
       </wsrf-rp:NewValues>
     </wsrf-rp:ResourcePropertyValueChangeNotification>
    </pr:Job_stateChangeNotification>

  </wsnt:Message>
 <wsnt:NotificationMessage>
</wsnt:Notify>
```

The `<wsrf-rp:ResourcePropertyValueChangeNotification>` element can be contained within an application-specific wrapper element. In this example, the print Job provides human readable information (job_name and printer_name) so that the notification message is also useful if sent directly to a user as notification that the print job is complete.

12. Glossary of Terminology

The following definitions outline the terminology and usage in this Primer.

Child:

- A sub-element in an XML document. Each child has a parent which contains it.

Resource:

- WSRF is concerned with resources which can be identified and described in XML to provide a view to requesters of their behaviour. In this sense, resources are things which exist in the implementation of an information system. These information system resources may represent any kind of logical or physical entities in the real world, (such as a logical shopping cart, or a physical printer) whose behaviour is being modelled by the system.

Resource property:

- A resource property is a piece of information defined as part of the description of a WS-Resource.
- A resource property may reflect a part of the resource's state.

Resource Properties document:

- The XML document representing a logical composition of resource property elements. The Resource Properties document defines a particular view or projection of the resource whose behaviour underlies the WS-Resource.

  The type (that is, the XML Schema definition of the root element) of a Resource Properties document is associated with the WSDL portType defining the Web service interface. This association is the basis of the WS-Resource definition. Each instance of a particular WS-Resource type implements a Resource Properties document of the type declared in the WSDL portType.

Resource property value element:

- The XML representation of a property of a WS-Resource instance. A resource property value element is a child of the root element of a resource properties instance document.

WS-Resource:

- A Web service which provides access to a resource and conforms to the constraints of the WS-Resource specification [WS-Resource].

13.References

[AppNotes]
WSRF Application Notes:
http://docs.oasis-open.org/wsrf/wsrf-application_notes-1.2-notes-cd-02.pdf

[RFC 2566]
The Internet Printing Protocol 1.0

[SOAP]
Simple Object Access Protocol (SOAP) 1.1
http://www.w3.org/TR/soap/

[URI]
T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC 2396, MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998.

[WS-Addressing]
Web Services Addressing 1.0 – Core:
http://www.w3.org/TR/ws-addr-core
Web Services Addressing 1.0 - WSDL Binding:
http://www.w3.org/TR/2006/WD-ws-addr-wsdl-20060216

[WS-BaseFaults]
http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf

[WS-BaseNotification]
http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-pr-02.pdf

[WS-Resource]
http://docs.oasis-open.org/wsrf/wsrf-ws_resource-1.2-spec-os.pdf

[WS-ResourceLifetime]
http://docs.oasis-open.org/wsrf/wsrf-ws_resource_lifetime-1.2-spec-os.pdf

[WS-ResourceProperties]
http://docs.oasis-open.org/wsrf/wsrf-ws_resource_properties-1.2-spec-os.pdf

[WS-ServiceGroup]
http://docs.oasis-open.org/wsrf/wsrf-ws_service_group-1.2-spec-os.pdf

[WSDL]
Web Services Description Language (WSDL) 1.1
http://www.w3.org/TR/wsdl

[XML Primer]
XML Schema Part 0: Primer Second Edition
http://www.w3.org/TR/xmlschema-0/

14.WSDL and Schema Files

The WSDL and schema files used in the examples can be found at http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-examples-cd-02.zip

This file contains:

- The simple shopping service in two versions:
  - For comparison with WSRF, a non-WSRF Web Services version defined by WSSimpleShoppingCart.wsdl with message definitions in WSSimpleShoppingCart.xsd and SimpleShoppingCart.xsd
  - A WSRF version defined by SimpleShoppingCart.wsdl and SimpleShoppingCart.xsd
- The Printer in two versions:
  - For comparison with WSRF, a non-WSRF Web services version defined by WSPrinter.wsdl, WSPrinter.xsd, and Printer.xsd
  - A WSRF version defined by Printer.wsdl and Printer.xsd
- The ShoppingCart, defined by ShoppingCart.wsdl and ShoppingCart.xsd.
- The PrinterAndJobGroup, defined by PrinterAndJobGroup.wsdl, PrinterAndJobGroup.xsd, and Printer.xsd

Appendix A Acknowledgments

Special thanks to the Global Grid Forum's Open Grid Services Infrastructure working group, which defined the OGSI v1.0 [OGSI] specification which was a large inspiration for the ideas expressed in the WSRF specifications.

The following individuals were members of the committee during the development of the WSRF specifications and this Primer:

Mario Antonioletti(EPCC, The University of Edinburgh), Akhil Arora (Sun Microsystems), Tim Banks (Author & Editor) (IBM), Jeff Bohren (OpenNetwork), Fred Carter (AmberPoint), Martin Chapman (Oracle), Glen Daniels (Sonic Software), David De Roure (University of Southampton), Thomas Freund (IBM), John Fuller (Individual), Stephen Graham (IBM), Anish Karmarkar (Oracle), Hideharu Kato (Hitachi), David Levine (IBM), Paul Lipton (Computer Associates), Mark Little (Arjuna Technologies Limited), Lily Liu (WebMethods, Inc.), Tom Maguire (IBM), Susan Malaika (IBM), David Martin (IBM), Samuel Meder (ArgonneNational Laboratory), Jeff Mischkinsky (Oracle), Roger Menday (Forschungszentrum Jlich GmbH), Bryan Murray (Hewlett-Packard), Mark Peel (Novell), Alain Regnier (Ricoh Company, Ltd.), Ian Robinson (IBM), Tom Rutt (Fujitsu), Matsunori Satomi (Hitachi), Igor Sedukhin (Computer Associates), Hitoshi Sekine (Ricoh Company, Ltd.), Frank Siebenlist (ArgonneNational Laboratory), Alex Sim (Lawrence Berkeley National Laboratory), David Snelling (Fujitsu), Latha Srinivasan (Hewlett-Packard), Jem Treadwell (Hewlett-Packard), Steve Tuecke (ArgonneNational Laboratory), William Vambenepe (Hewlett-Packard), Katy Warr (IBM), Alan Weissberger (NEC Corporation), Pete Wenzel (SeeBeyond Technology Corporation), Kirk Wilson (Computer Associates) and Umit Yalcinalp (SAP).

In addition, the following people made contributions by reviewing and discussing the contents:

Roland Merrick (IBM), Martin Gale (IBM), Susan Malaika (IBM), Tom Cain (IBM), Alain Reigner (Ricoh Company, Ltd.), Simon Laws (IBM), Roger Menday (Forschungszentrum Jlich GmbH), Kirk Wilson (Computer Associates), Jem Treadwell (HP), Ian Robinson (IBM).

Appendix B 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.

Appendix C Revision History

Rev	Date	By Whom	What
Drafts 9	2005-11-22	Tim Banks	• Added 'Notices' section • Corrected namespace prefixes in examples (ws-ssc -> ssc etc). • Removed interface-semi-circles from diagram in section 10 (printer and Jobs) • Improved section 3: 'WSRF in a nutshell' to mention all aspects of the specs. • Addressed comments from Kirk Wilson (introduction + improved terminology around 'resource') • Incorporate review comments by Jem Treadwell • Incorporated review comments by Ian Robinson: restructured section 11 (ServiceGroup)
Draft wd-10	2005-11-24	Tim Banks	• Revised section 3 (WSRF in a Nutshell) to include examples of EPRs • Rationalised EPRs and resynched with xml examples.
Draft cd-01	2005-11-24	Tim Banks	• Updated pointer to wsdl and xml files • Updated title page & status
Draft wd-12	2006-02-20	Tim Banks	• Added Hyperlinks to example source files.
Draft wd-14	2006-03-09	Tim Banks	• Improved Hyperlinks, Converted to OpenDocument Format
Draft wd-15	2006-05-02	Tim Banks	• Review comments from Ian Robinson and Jem Treadwell.
Draft cd-02	2006-05-23	Tim Banks	• Updated title page & status