



29 to and send comments to the [wsrf-comment@lists.oasis-open.org](mailto:wsrf-comment@lists.oasis-open.org) list. To subscribe, send an email  
30 message to [wsrf-comment-request@lists.oasis-open.org](mailto:wsrf-comment-request@lists.oasis-open.org) with the word "subscribe" as the body of  
31 the message.

32  
33 For information on whether any patents have been disclosed that may be essential to implementing  
34 this specification, and any offers of patent licensing terms, please refer to the Intellectual Property  
35 Rights section of the WSRF TC web page (<http://www.oasis-open.org/committees/wsrf/>).

## Table of Contents

37	<b>1 INTRODUCTION .....</b>	<b>5</b>
38	1.1 GOALS AND REQUIREMENTS .....	6
39	1.1.1 Requirements .....	6
40	1.1.2 Non-Goals.....	6
41	1.2 TERMINOLOGY .....	7
42	1.3 NAMESPACES .....	8
43	<b>2 TERMINOLOGY AND CONCEPTS.....</b>	<b>9</b>
44	<b>3 EXAMPLE .....</b>	<b>10</b>
45	3.1 THE OPERATINGSYSTEM PORTTYPE.....	10
46	3.2 OPERATING SYSTEM PROPERTIES.....	11
47	3.2.1 Operating System Resource Property definitions .....	11
48	3.2.2 Identification Property definitions.....	11
49	3.2.3 MetadataDescriptor for Identification portType .....	12
50	3.2.4 MetadataDescriptor for OperatingSystem portType .....	13
51	<b>4 LOGICAL MODEL FOR METADATA.....</b>	<b>16</b>
52	<b>5 INFORMATION MODEL FOR WS-RESOURCE METADATA .....</b>	<b>17</b>
53	<b>6 DEFINITIONS COMPONENT .....</b>	<b>18</b>
54	6.1 METADATADESCRIPTOR COMPONENTS WITHIN A DEFINITIONS COMPONENT .....	19
55	<b>7 METADATADESCRIPTOR COMPONENT .....</b>	<b>20</b>
56	7.1 PROPERTIES COMPONENT OF A METADATADESCRIPTOR .....	21
57	<b>8 PROPERTY COMPONENT .....</b>	<b>22</b>
58	8.1 XML SCHEMA VALUE SPACE AND {VALIDVALUES} .....	24
59	8.2 VALIDVALUES .....	25
60	8.3 VALIDVALUERANGE.....	26
61	8.4 STATICVALUES .....	28
62	8.5 INITIALVALUES.....	29
63	<b>9 DOCUMENTATION COMPONENT .....</b>	<b>31</b>
64	<b>10 OBTAINING A METADATADESCRIPTOR DOCUMENT .....</b>	<b>31</b>
65	10.1 EXTENDING WSDL 1.1 PORTTYPE.....	31
66	10.2 USING RESOURE PROPERTY ELEMENTS TO EXPOSE METADATADESCRIPTORS.....	32
67	<b>11 REFERENCES .....</b>	<b>34</b>

68	11.1	NORMATIVE .....	34
69	11.2	NON-NORMATIVE.....	34
70		<b>APPENDIX A. ACKNOWLEDGMENTS .....</b>	<b>35</b>
71		<b>APPENDIX B. XML SCHEMA FOR WS-RESOURCEMETADATADESCRIPTOR.....</b>	<b>36</b>
72		<b>APPENDIX C. REVISION HISTORY.....</b>	<b>45</b>
73		<b>APPENDIX D. NOTICES.....</b>	<b>46</b>
74			

## 1 Introduction

In the WS-Resource Framework [WSRF], elements of a WS-Resource's state are exposed to third party requestors through an XML document. The XML document associated with a WS-Resource is called a *resource properties document*. The resource properties document is a projection of the WS-Resource's state (not all of the elements of a WS-Resource's state are exposed through the resource properties document). An individual element of state contained in a resource properties document is called a *resource property*. Access to the resource properties document is governed by Web services operations defined in the WS-Resource Properties [WS-Resource Properties] specification. These operations generically allow for get, set and query of resource properties.

In many cases, some of the resource properties exposed through the resource properties document are not accessible through every operation defined in WS-ResourceProperties. The most common case of this is a resource property that is "read-only" implying that a requestor may not use Web services message exchanges (such as the WS-ResourceProperties SetResourceProperties operation) to change the value of the resource property. Clearly, an implementation of a WS-Resource is likely to return a fault message if a requestor attempts to change the value of a "read-only" resource property. However, in the absence of additional metadata, there is no standard means by which the requestor can determine a priori that the resource property was not modifiable.

We refer to the concept of a WS-Resource Metadata Descriptor to describe a unit of metadata information associated with the interface components of a WS-Resource. We describe an information model that outlines the components of metadata and their relationships to interface description artifacts such as WSDL 1.1 portTypes and resource properties document schema definitions.

A WS-Resource Metadata Descriptor serves multiple purposes. The first is to provide additional information about the resource properties of a WS-Resource. For instance, indicating whether a resource property is changeable using Web services message exchanges such as the SetResourceProperties operation described in the WS-ResourceProperties specification [WS-ResourceProperties]. This aspect of the MetadataDescriptor is associated with the interface of the WS-Resource, and would not vary between different implementations of the interface. Information in the MetadataDescriptor provides clients of a WS-Resource the potential for greater understanding of the behavior of that WS-Resource.

The second purpose is to provide information about the value restrictions of the resource properties in the resource properties document for the WS-Resource. This additional information may be associated with implementations of the interface as well as with the WSDL interface definition.

115 The single portType that describes the manageability interface for a manageable resource type is  
116 derived from various other manageability portTypes. With WSDL 1.1, physically including, using  
117 copy and paste, the operations from each of these portTypes into the definition of the most  
118 derived portType achieves this inheritance. Each of the portTypes from which a manageable  
119 resource's portType is derived may also have a MetadataDescriptor to augment its description.  
120 Each of the portTypes may have an optional attribute information item that references a  
121 MetadataDescriptor component by its namespace qualified name (QName).

122

123 This document standardizes the form of the WS-Resource MetadataDescriptor that contains  
124 metadata information about a WS-Resource's interface so that clients of that interface may  
125 reason about implementations of the interface at both design time and run time. The syntax of a  
126 preferred XML serialization of the information model is also described.

127

128 A portion of the Global Grid Forum's "Open Grid Services Infrastructure (OGSI) Version 1.0"  
129 specification [OGSI] inspired many of the concepts expressed in this document.

130

## 131 **1.1 Goals and Requirements**

132 The goal of this document is to define the terminology, concepts, information model and XML  
133 definitions needed to express the metadata requirements of WS-Resources, as defined by the  
134 [WS-Resource] specification.

135

### 136 **1.1.1 Requirements**

137 In meeting this goal, the specification MUST address the following specific requirements:

- 138 • Define an information model representing metadata about resource properties associated  
139 with a WS-Resource interface.
- 140 • Define a standard annotation for associating metadata descriptions with other description  
141 artifacts of the WS-Resource, particularly its WSDL 1.1 portType and its resource  
142 properties document definition.
- 143 • Define the standard schema for representing the aspects of the information model.

144

### 145 **1.1.2 Non-Goals**

146 The following topics are outside the scope of this specification:

- 147 • It is not an objective of this specification to define new message exchanges required to  
148 access the metadata from a WS-Resource.
- 149 • It is not an objective of this specification to describe the means required to store the  
150 metadata for a WS-Resource.

## 151 1.2 Terminology

152 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",  
153 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be  
154 interpreted as described in [RFC 2119].

155

156 When describing abstract data models, this specification uses the notational convention used by  
157 the [XML Infoset]. Specifically, abstract property names always appear in square brackets (e.g.,  
158 [some property]).

159

160 This specification uses a notational convention, referred to as "Pseudo-schemas" in a fashion  
161 similar to the WSDL 2.0 Part 1 specification [WSDL 2.0]. A Pseudo-schema uses a BNF-style  
162 convention to describe attributes and elements:

163 '?' denotes optionality (i.e. zero or one occurrences),

164 '\*' denotes zero or more occurrences,

165 '+' one or more occurrences,

166 '[' and ']' are used to form groups,

167 '|' represents choice.

168

169 Attributes are conventionally assigned a value which corresponds to their type, as defined in the  
170 normative schema.

171

```
172 <!-- sample pseudo-schema -->  
173 <element  
174     required_attribute_of_type_QName="xs:QName"  
175     optional_attribute_of_type_string="xs:string"? >  
176     <required_element />  
177     <optional_element />?  
178     <one_or_more_of_these_elements />+  
179     [ <choice_1 /> | <choice_2 /> ]*  
180 </element>
```

181

182

183

184

185

186

187 **1.3 Namespaces**

188 The following namespaces are used in this document:

Prefix	Namespace
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>
wsdl	<a href="http://schemas.xmlsoap.org/wsdl">http://schemas.xmlsoap.org/wsdl</a>
wsrf-rp	<a href="http://docs.oasis-open.org/wsrp/rp-2">http://docs.oasis-open.org/wsrp/rp-2</a>
wsrmd	<a href="http://docs.oasis-open.org/wsrp/rmd-1">http://docs.oasis-open.org/wsrp/rmd-1</a>

189



## 190 **2 Terminology and Concepts**

191 The following definitions outline the terminology and usage in this specification. This section gives  
192 only brief description of these terms.

193

194 **Metadata:**

- 195 • Data about data. In practice, metadata comprises a structured set of descriptive elements  
196 to describe an information resource. Currently, only a WS-Resource's resource  
197 properties have metadata definitions.

198

199 **MetadataDescriptor:**

- 200 • A unit of containment for resource property metadata for a WS-Resource's interface;  
201 property metadata is defined by zero or more Property elements.

202

203 **MetadataDescriptor Document:**

- 204 • An XML instance document whose root is a Definitions element from the wsrmd  
205 namespace. This document contains definitions for zero or more MetadataDescriptor  
206 components.

## 207 **3 Example**

208 In the following example there are “Operating System” WS-Resources whose values are  
209 projected from the implementation of the *OperatingSystem* portType.

### 210 **3.1 The OperatingSystem portType**

211 The *OperatingSystem* portType defines operations and a resource properties document that  
212 describes the Web services interface to operating system resource instances. As well as  
213 providing a mechanism for interacting with the operating system itself, this portType also  
214 describes properties, which represent the hardware of the underlying machine on which the  
215 operating system is running.

216

217 The *OperatingSystem* portType is derived from various other manageability portTypes – these  
218 illustrations use some of the function from the *Identification* manageability portType. As is  
219 required with WSDL 1.1, this derivation is achieved by physically including the definitions from  
220 each of these portTypes in the definition of the *OperatingSystem* portType. This “cut-and-paste”  
221 system of derivation is discussed further in [AppNotes].

222

223 The *OperatingSystem* portType is sketched as follows:

224

```
225 (01) ... xmlns:os="http://example.com/ns/OperatingSystem"  
226 (02) ... xmlns:id="http://example.com/ns/Identification"  
227 (03) ...  
228 (04) <portType name="OperatingSystem"  
229 (05)   wsrf-rp:ResourceProperties="os:OSResourceProperties"  
230 (06)   ..wsrmd:Descriptor="os:OperatingSystemMetadataDescriptor"  
231 (07)   ..wsrmd:DescriptorLocation="http://example.com/OperatingSystem.wsrmd" >  
232 (08) ...  
233 (09) </portType>
```

234

235 Line (04) contains a portType declaration for a portType named *OperatingSystem* in the  
236 namespace corresponding to the *os:* namespace prefix declaration.

237

238 Line (05) indicates the global XML element declaration of the root element of the resource  
239 properties document associated with any WS-Resource whose Web service implements the  
240 *os:OperatingSystem* portType.

241

242 Line (06) identifies that a *MetadataDescriptor* has been defined for this interface identified by the  
243 QName *os:OperatingSystemMetadataDescriptor*.

244

245 Line (07) - (07) indicate that information about MetadataDescriptors in the namespace  
246 corresponding to the os: namespace prefix declaration can be found by dereferencing  
247 <http://example.com/OperatingSystem.wsrmd>.

## 248 3.2 Operating System Properties

249 In the following example we define a subset of the resource properties of the operating system.  
250 Following that are the MetadataDescriptors for the Identification and Operating System  
251 portTypes.

### 252 3.2.1 Operating System Resource Property definitions

```
253 (10) ...  
254 (11)   xmlns:os="...  
255 (12)  
256 (13) <element name="OSResourceProperties">  
257 (14)   <complexType>  
258 (15)     <sequence>  
259 (16)       <element ref="id:ResourceType" minOccurs="0" maxOccurs="1"/>  
260 (17)       <element ref="id:ResourceID" minOccurs="0" maxOccurs="1"/>  
261 (18)       <element ref="os:numberOfProcesses" minOccurs="0" maxOccurs="1"/>  
262 (19)       <element ref="os:totalSwapSpaceSize" minOccurs="0" maxOccurs="1"/>  
263 (20)       <element ref="os:processor" minOccurs="1" maxOccurs="unbounded"/>  
264 (21)     </sequence>  
265 (22)   </complexType>  
266 (23) </element>  
267 (24)  
268 (25) <element name="numberOfProcesses" type="xsd:int" />  
269 (26) <element name="totalSwapSpaceSize" type="xsd:unsignedLong" />  
270 (27) <element name="processor" type="xsd:string" />  
271 (28)
```

### 272 3.2.2 Identification Property definitions

273

```
274 (29) <element name="IdentificationResourceProperties">  
275 (30)   <complexType>  
276 (31)     <sequence>  
277 (32)       <element ref="ResourceType" minOccurs="0" maxOccurs="1"/>  
278 (33)       <element ref="ResourceID" minOccurs="0" maxOccurs="1"/>  
279 (34)     </sequence>  
280 (35)   </complexType>  
281 (36) </element>
```

282  
283  
284

```
(37)  
(38) <element name="ResourceType" type="xs:string" />  
(39) <element name="ResourceID" type="xs:string" />
```

### 285 3.2.3 MetadataDescriptor for Identification portType

286 An example MetadataDescriptor document for the Identification portType is included below. For  
287 the purposes of this example, this MetadataDescriptor document will be located at the URL  
288 <http://example.com/metadataDescriptors/Identification.wsrm>.

289

290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307

```
(40)  
(41) <Definitions  
(42)     xmlns="http://docs.oasis-open.org/wsr/1.0/"  
(43)     xmlns:id="http://example.com/ns/Identification"  
(44)     targetNamespace="http://example.com/ns/Identification">  
(45)   <MetadataDescriptor  
(46)     name="IdentificationMetadataDescriptor"  
(47)     interface="id:Identification"  
(48)     wsdlLocation="http://example.com/ns/Identification  
(49)       http://example.com/wsd/Identification.wsdl" >  
(50)   <Property name="id:ResourceID"  
(51)     mutability="constant"  
(52)     modifiability="read-only" />  
(53)   <Property name="id:ResourceType"  
(54)     mutability="constant"  
(55)     modifiability="read-only" />  
(56)   </MetadataDescriptor>  
(57) </Definitions>
```

308

309 Line (41) contains a Definitions element defining MetadataDescriptor elements for the target  
310 namespace identified in line (44).

311

312 There is one MetadataDescriptor element child of this Definitions element (lines (45)–(56)).

313 The name of the MetadataDescriptor is contained in line (46). This together with the namespace  
314 prefix declaration in line (43) corresponding to the targetNamespace of the Definitions element  
315 means the QName of the MetadataDescriptor is id:IdentificationMetadataDescriptor.

316

317 Line (47) identifies the QName of the portType (interface) with which this MetadataDescriptor is  
318 associated. The location of WSDL for the Identification portType is expressed in the wsdlLocation  
319 attribute in lines (48)–(49). This follows the pattern of the wsdl:wsdlLocation attribute defined in  
320 the WSDL 2.0 specification [WSDL 2.0].

321

322 Lines (50)-(55) show two Property elements containing metadata information about resource  
323 properties defined in the resource properties document for the Identification portType. Lines (50)-  
324 (52) contain the first Property element that references the QName of the id:ResourceID resource  
325 property. Line (51) indicates that the id:ResourceID resource property element will always have a  
326 constant value. Line (52) states that the id:ResourceID resource property is read-only, meaning  
327 that it cannot be changed by a requestor using Web services message exchanges such as the  
328 SetResourceProperties operation as defined in WS-ResourceProperties.

329

330 The second Property element, in lines (53)-(55), references the QName of the id:ResourceType  
331 resource property in line (53). This resource property element has the same metadata attributes  
332 as id:ResourceIdentifier.

### 333 **3.2.4 MetadataDescriptor for OperatingSystem portType**

334 For the purposes of this example the MetadataDescriptor document for the OperatingSystem  
335 portType is found at <http://example.com/metadataDescriptors/OperatingSystem.wsrm>.  
336

337 The contents of the MetadataDescriptor for the OperatingSystem portType appear as follows:

338

```
339 (58)      <Definitions
340 (59)          xmlns="http://docs.oasis-open.org/wsrf/rmd-1"
341 (60)          xmlns:id="http://example.com/ns/Identification"
342 (61)          xmlns:os="http://example.com/ns/OperatingSystem"
343 (62)          targetNamespace="http://example.com/ns/OperatingSystem">
344 (63)      <MetadataDescriptor
345 (64)          name="OperatingSystemMetadataDescriptor"
346 (65)          interface="os:OperatingSystem"
347 (66)          wsdlLocation="http://example.com/ns/OperatingSystem
348 (67)              http://example.com/wsdl/OperatingSystem.wsdl">
349 (68)      <Property name="id:ResourceID"
350 (69)          mutability="constant"
351 (70)          modifiability="read-only" />
352 (71)      <Property name="id:ResourceType"
353 (72)          mutability="constant"
354 (73)          modifiability="read-only">
355 (74)          <ValidValues>
356 (75)              <id:ResourceType>SuSELinux</id:ResourceType>
357 (76)              <id:ResourceType>IBMzOS</id:ResourceType>
358 (77)              <id:ResourceType>MicrosoftWindows_XP</id:ResourceType>
359 (78)          </ValidValues>
360 (79)      </Property>
```

```

361 (80)         <Property name="os:numberOfProcesses"
362 (81)             mutability="mutable"
363 (82)             modifiability="read-only" />
364 (83)         <Property name="os:processor"
365 (84)             mutability="constant"
366 (85)             modifiability="read-only">
367 (86)             <ValidValues>
368 (87)                 <os:processor>Pentium Family</os:processor>
369 (88)                 <os:processor>Power PC 750</os:processor>
370 (89)                 <os:processor>68xxxx Family</os:processor>
371 (90)                 <os:processor>PA-RISC Family</os:processor>
372 (91)                 <os:processor>Alpha Family</os:processor>
373 (92)                 <os:processor>IBM390 Family</os:processor>"
374 (93)                 <os:processor>G5</os:processor>
375 (94)                 <os:processor>AMD</os:processor>
376 (95)             </ValidValues>
377 (96)         </Property>
378 (97)     </MetadataDescriptor>
379 (98) </Definitions>

```

380 Lines (58) to (98) contain a Definitions element for the <http://example.com/ns/OperatingSystem>  
381 namespace.

382

383 Lines (63) to (98) contain the definition of the MetadataDescriptor with QName  
384 os:OperatingSystem MetadataDescriptor.

385

386 Line (65) indicates that this MetadataDescriptor corresponds to the OperatingSystem portType.

387 Lines (66)-(67) gives the location of the WSDL document that defines elements associated with  
388 the namespace URI associated with the os: prefix (i.e. the WSDL definitions element that defines  
389 the OperatingSystem portType).

390

391 Lines (68)-(96) contains the four properties described in this MetadataDescriptor example.

392 Lines (68) –(70) contain the ResourceID Property element copied from the  
393 id:IdentificationMetadataDescriptor describing the Identification portType.

394 Lines (71)-(79) contain the Property element describing the id:ResourceType resource property  
395 from the Identification portType. Lines (74)-(78) contain the set of ValidValues that the  
396 id:ResourceType resource property may contain.

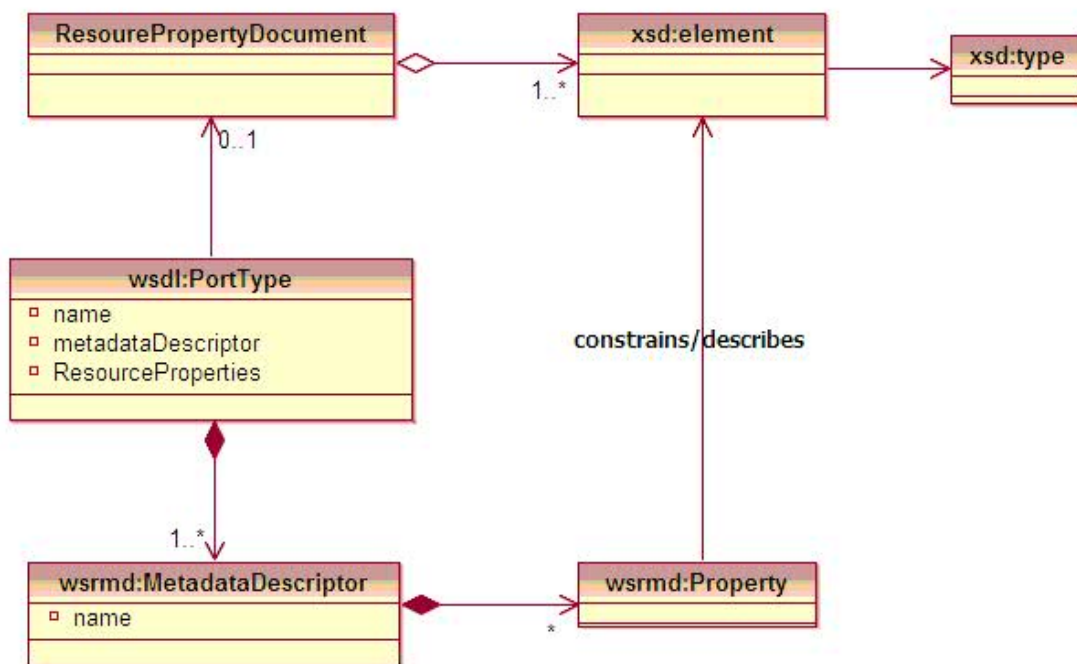
397

398 Lines (80)-(82) contain the os:numberOfProcesses Property element which references the  
399 QName of the os:numberOfProcesses resource property. Line (81) indicates that the value of the  
400 os:numberOfProcesses may change over time. Line (82) indicates that, the

401 os:numberOfProcesses can not be changed by a requestor using Web services message  
402 exchanges such as the SetResourceProperties operation as defined in WS-ResourceProperties  
403 [WS-ResourceProperties].  
404 The next Property element references the os:processor. The modifiability and mutability values  
405 indicate that the property is static – it will not change during the resource’s lifetime. Lines (87)-  
406 (94) describe valid values for the os:processor.

407 **4 Logical Model for Metadata**

408 The following figure shows a logical model depicting the relationship between the various  
 409 elements of metadata description and those elements the metadata describes.



410

411

412

Figure 1 Logical Model of WS-Resource MetadataDescriptor

413 In our model, the unit of metadata containment is referred to as a *MetadataDescriptor*. A  
 414 *MetadataDescriptor* is used to describe aspects of a WS-Resource's interface, particularly those  
 415 elements associated with the WS-Resource's WSDL 1.1 portType.

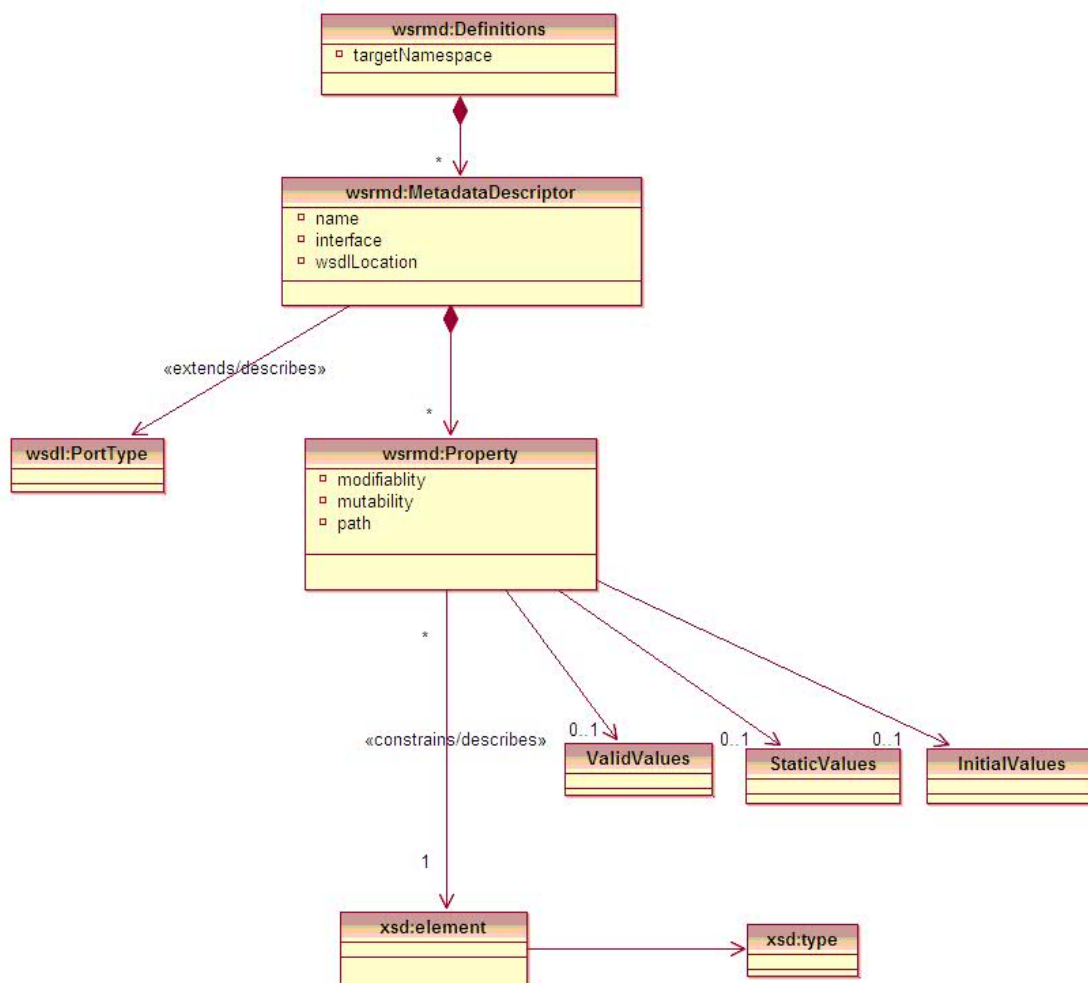
416

417 A *MetadataDescriptor* contains metadata describing and/or constraining resource property  
 418 elements as contained within a WS-Resource's Resource Properties document type definition.  
 419 Each resource property element is defined as an XML Schema global element, in some  
 420 namespace.



421 **5 Information Model for WS-Resource Metadata**

422 This section describes the information model for metadata describing/constraining the resource  
 423 properties of WS-Resources. The model is a simple hierarchy – each WSDL portType *references*  
 424 a Resource Metadata Descriptor document, and that file *contains* a Definitions element, which  
 425 *contains* MetadataDescriptors, which *contain* Property elements. A UML diagram of this model is  
 426 shown in the following figure:



427

428 *Figure 2 - Information Model for WS-Resource Metadata Descriptor*

429

430 We describe the Definitions, MetadataDescriptor, and Property components in the following  
 431 sections.

## 6 Definitions Component

The Definitions component is a container for a set of MetadataDescriptor components (see section 7). The Definitions component defines a targetNamespace which forms the {namespace} property of all components it contains.

The properties of a Definitions component are as follows:

- {targetNamespace} a namespace URI that applies as the {namespace} property to all [child] components.
- {metadataDescriptors} a set of zero or more MetadataDescriptor components.

The following is an XML representation of the Definitions component:

```
<Definitions
  targetNamespace="xs:anyURI"
  {anyAttribute}* >

  <documentation />?
  <MetadataDescriptor /> *

</Definitions>
```

The Definitions *element information item* has the following Infoset properties:

- A [local name] of "Definitions".
- A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".
- one or more *attribute information items* amongst its [attributes] as follows:
  - A REQUIRED targetNamespace *attribute information item*
    - The value of this attribute information item contains a URI that defines the {namespace} property of all [child] components.
    - The type of the targetNamespace *attribute information item* is xs:anyURI.
  - Zero or more namespace qualified *attribute information items*. The [namespace name] of such *attribute information items* MUST NOT be "http://docs.oasis-open.org/wsrf/rmd-1".
- Zero or more *element information items* amongst its [children], in order as follows:
  - An OPTIONAL documentation *element information item* (See section 7).
  - Zero or more MetadataDescriptor *element information items* (See section 6.1).

469 **6.1 MetadataDescriptor components within a Definitions**  
470 **component**

471 All MetadataDescriptor components defined in a given namespace MUST appear as [children] of  
472 a Definitions component with {targetNamespace} value the same URI as that namespace. All  
473 MetadataDescriptor components MUST be uniquely named, implying that the {name} property of  
474 the MetadataDescriptor component MUST be unique amongst the {metadataDescriptors} of a  
475 Definitions component.

476

## 7 MetadataDescriptor Component

477

The MetadataDescriptor component is a container for a set of metadata descriptions and constraints on a WS-Resource. The MetadataDescriptor component contains additional information that describes or constrains various aspects of a WS-Resource. For example, it provides additional information about the interface of the WS-Resource relevant to the management of the resource. In particular, it allows tools and applications, such as management applications, to be able to reason in detail about the WS-Resource both at runtime and at development time when no instances of the WS-Resource are available.

484

485 The properties of a MetadataDescriptor component are as follows:

486

- {name} a name of a MetadataDescriptor component.
- {namespace} a namespace URI of the MetadataDescriptor component.
- {QName} a combination of the {name} and {namespace} of the MetadataDescriptor component.
- {interface} a QName identifying a Web services interface definition with which this MetadataDescriptor is associated.
- {wsdlLocation} a set of URI pairs, each pair associating a namespace URI with a URL of a document containing a WSDL definition of that namespace. This is a similar mechanism to that used in WSDL 2.0 [WSDL2.0].
- {properties} A set of zero or more Property components

487

488

489

490

491

492

493

494

495

496

497 The following is an XML representation of the MetadataDescriptor component:

498

499

```
<MetadataDescriptor
  name="xs:NCName"
  interface="xs:QName"
  wsdlLocation="list of xs:anyUri"?
  {anyAttribute}* >
  <documentation /> ?
  <Property /> *
  {any}*
</MetadataDescriptor>
```

500

501

502

503

504

505

506

507

508

509

510 The MetadataDescriptor *element information item* has the following Infoset properties:

511

- A [local name] of "MetadataDescriptor".
- A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".

512

- 513 • two or more *attribute information items* amongst its [attributes] as follows:
  - 514 ○ A REQUIRED name *attribute information item*
    - 515 ▪ The value of this *attribute information item* contains the name of this
    - 516 MetadataDescriptor component.
    - 517 ▪ The type of the name *attribute information item* is xs:NCName.
  - 518 ○ A REQUIRED interface *attribute information item*
    - 519 ▪ The value of this *attribute information item* contains a QName of a WSDL
    - 520 1.1 portType element or WSDL 2.0 interface element associated with this
    - 521 MetadataDescriptor component.
    - 522 ▪ The type of the interface *attribute information item* is xs:QName.
  - 523 ○ An OPTIONAL wsdlLocation *attribute information item*
    - 524 ▪ The value of this *attribute information item* contains a list of pairs of
    - 525 URIs; where the first URI of the pair, which MUST be an absolute URI as
    - 526 defined in [URI], indicates a WSDL namespace name, and, the second a
    - 527 hint as to the location of a WSDL document defining WSDL components
    - 528 for that namespace name. The second URI of a pair MAY be absolute or
    - 529 relative.
    - 530 ▪ The type of the wsdlLocation *attribute information item* is list of
    - 531 xs:anyURI.
  - 532 ○ Zero or more namespace qualified *attribute information items*. The [namespace
  - 533 name] of such *attribute information items* MUST NOT be "http://docs.oasis-
  - 534 open.org/wsrf/rmd-1".
- 535 • Zero or more *element information items* amongst its [children], in order as follows:
  - 536 ○ An OPTIONAL documentation *element information item* (See section 9).
  - 537 ○ Zero or more Property *element information items* (See section 8)
  - 538 ○ Zero or more namespace-qualified *element information items*. The [namespace
  - 539 name] of such *element information items* MUST NOT be "http://docs.oasis-
  - 540 open.org/wsrf/rmd-1".

## 541 7.1 Properties component of a MetadataDescriptor

542 The {properties} of a MetadataDescriptor contains a set of Property components, defining  
543 additional metadata and constraints on resource property elements (and attributes) associated  
544 with a MetadataDescriptor. The definition of a Property component's scope is contained in  
545 Section **Error! Reference source not found.**

546

## 8 Property Component

547

The Property component is a container for a set of metadata descriptions and constraints on a specific Resource Property element or attribute thereof. The properties of a Property component are as follows:

548

549

550

- {name} an identifier of the XML element to which the Property component applies. This is defined as a resource property's QName.

551

552

- {mutability} an xs:string enumeration of "constant", "appendable", or "mutable".

553

- {modifiability} an xs:string enumeration of "read-only" or "read-write".

554

- {subscribability} an xs:boolean indicating, if true, that the Resource Property element associated with the {name} can be the target of a subscription.

555

556

- {validValues} optional choice of one ValidValues component or one ValidValueRange component.

557

558

- {staticValues} optional choice of one StaticValues component.

559

- {initialValues} optional choice of one InitialValues component.

560

- {attributes} zero or more Attribute components.

561

562

A Property component MAY also contain additional "extension" components added using the extensibility mechanism defined by this specification. The following is an XML representation of the Property component:

563

564

565

566

```
<Property
  name="xs:QName"
  mutability="[constant|appendable|mutable]" ?
  modifiability="[read-only|read-write]" ?
  subscribability="xs:boolean" ?
  {anyAttribute}* >

  <documentation />?
  [ <ValidValues> {any}* </ValidValues> |
    <ValidValueRange
      lowerBound="xs:anySimpleType"? upperBound="xs:anySimpleType"?
    /> ] ?
  <StaticValues> {any}* </StaticValues> ?
  <InitialValues> {any}* </InitialValues> ?
  {any}*
</Property>
```

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

The Property *element information item* has the following Infoset properties:

- 584 • A [local name] of "Property" .
- 585 • A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".
- 586 • one or more *attribute information items* amongst its [attributes] as follows:
  - 587 ○ A REQUIRED name attribute information item
    - 588 ▪ The value of this *attribute information item* MUST contain a QName of
    - 589 the Resource Property (an XML Schema global element definition)
    - 590 contained within the Resource Properties document associated with the
    - 591 portType or interface identified by {interface}. The Resource Property
    - 592 element MUST conform to the requirements specified for Resource
    - 593 Property declarations in WS-ResourceProperties.
  - 594 ○ An OPTIONAL mutability attribute information item
    - 595 ▪ The value of this *attribute information item* expresses how the value of
    - 596 the {name} can change over time.
    - 597 ▪ The type of the mutability *attribute information item* is an xs:string
    - 598 restricted to the following enumeration:
      - 599 • "constant"
        - 600 The values of the {name} MUST NOT change after WS-
        - 601 Resource creation.
      - 602 • "mutable"
        - 603 The values of the {name} MAY change at any time during
        - 604 the lifetime of the WS-Resource. Existing values MAY be
        - 605 removed and new values MAY be added.
      - 606 • "appendable"
        - 607 The values of the {name} MAY have new values added
        - 608 during the lifetime of the WS-Resource. Once added those
        - 609 values MUST NOT be removed.
    - 610 ▪ If the mutability *attribute information item* is not defined, the value of the
    - 611 mutability property is "unknown".
  - 612 ○ An OPTIONAL modifiability *attribute information item*
    - 613 ▪ The value of this *attribute information item* indicates whether a requestor
    - 614 can modify the value of the {name}.
    - 615 ▪ The type of the modifiability *attribute information item* is an xs:string
    - 616 restricted to the following enumeration:
      - 617 • "read-only" – The value of the {name} can not be changed by
      - 618 Web services message exchanges such as the
      - 619 SetResourceProperty message as defined in WS-
      - 620 ResourceProperties.
      - 621 • "read-write" – The value of the {name} MAY be changed by Web
      - 622 services message exchanges such as the SetResourceProperty

- 623 message as defined in WS-ResourceProperties. Note - If the  
624 value of the modifiability *attribute information item* is “read-write”  
625 then the value of the mutability *attribute information item* MUST  
626 NOT be “constant”.
- 627     ▪ If the modifiability attribute information item is not defined, the value of  
628 the modifiability property is “unknown”.
  - 629     ○ An OPTIONAL subscribability *attribute information item*
    - 630         ▪ The value of this *attribute information item* expresses whether the  
631 Resource Property element associated with the {name} can be the target  
632 of a subscription. The default value is “false”. Note: The actual  
633 subscription semantics are dependent on whatever notification  
634 mechanism, if any, (such as WS-BaseNotification [WS-BaseNotification])  
635 is supported.
    - 636         ○ Zero or more namespace qualified *attribute information items*. The [namespace  
637 name] of such *attribute information items* MUST NOT be "http://docs.oasis-  
638 open.org/wsrf/rmd-1".
  - 639     • Zero or more *element information items* amongst its [children], in order as follows:
    - 640         ○ An OPTIONAL documentation *element information item* (See section 9).
    - 641         ○ An OPTIONAL *element information item* from among the following:
      - 642             ○ A ValidValues element information item (See section 8.2)
      - 643             ○ A ValidValueRange element information item (See section 8.3)
      - 644             ○ An OPTIONAL StaticValues element information item (See section 8.4)
      - 645             ○ An OPTIONAL InitialValues element information item (See section 8.5)
    - 646         ○ Zero or more namespace-qualified *element information items*. The [namespace  
647 name] of such *element information items* MUST NOT be "http://docs.oasis-  
648 open.org/wsrf/rmd-1".

## 649 **8.1 XML Schema value space and {validValues}**

650 When creating a resource property (ie defining an XML Global Element), the XML Schema  
651 designer defines the semantic of the property and uses XML Schema to express the value space  
652 of the resource property (based on the semantics of the property) and all of its descendant  
653 *element information items* and *attribute information items*. This is a different concept from what is  
654 expressed by defining the {validValues}. When specifying {validValues} in a metadata description,  
655 one does not redefine the semantic of the {name} nor its value space. Specifying {validValues}  
656 expresses constraints on the value space that are appropriate for the specific use of the {name}.  
657 This distinction should guide designers in deciding whether to use XML Schema mechanisms or  
658 a metadata description to restrict value space of a {name}. The value space defined by  
659 {validValues} for a {name} MUST be contained within the XML Schema definition of the {name}.



## 660 8.2 ValidValues

661 The purpose of the ValidValues component is to restrict the set of valid values that a [parent]  
662 Property component's {name} may contain.

663

664 If the {validValues} of a Property component is not empty, and contains a ValidValues description,  
665 then any Web service that implements the portType or interface identified by {interface} MUST  
666 ensure that the value(s) of the {name} of the [parent] Property component MUST correspond to  
667 one of the values enumerated within the set of {validValues}.

668

669 Note: because the child *element information items* of a ValidValues *element information item* are  
670 XML fragments, it is not required that these fragments be validated (processContents is "skip"). .

671 The properties of a ValidValues component are as follows.

672

- 673 • {values} zero or more XML fragments that correspond to the type of the [parent] Property  
674 component's {name}.

675

676 The following is an XML representation of the ValidValues component:

677

```
678 <ValidValues  
679     {anyAttribute}* >  
680     <documentation />?  
681     {any}*  
682 </ValidValues>
```

683

684 The ValidValues *element information item* has the following Infoset properties:

- 685 • A [local name] of "ValidValues" .
- 686 • A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".
- 687 • zero or more *attribute information items* amongst its [attributes] as follows:
  - 688 ○ Zero or more namespace qualified *attribute information items*. The [namespace  
689 name] of such *attribute information items* MUST NOT be "http://docs.oasis-  
690 open.org/wsrf/rmd-1".
- 691 • Zero or more *element information items* amongst its [children], either:
  - 692 ○ An OPTIONAL documentation *element information item* (See section 9).
  - 693 ○ Zero or more namespace-qualified *element information items*. The [namespace  
694 name] of such *element information items* MUST NOT be "http://docs.oasis-  
695 open.org/wsrf/rmd-1".

- 696
- 697
- 698
- 699
- 700
- 701
- 702
- 703
- Each *element information item* MUST be an XML fragment that corresponds to the type of the XML element identified by the [parent] Property component's {name}
  - Note, because these are XML fragments, it is not expected that a processor of a MetadataDescriptor document would need to validate these element information items (processContents = "skip").
- Zero or more character information items.

### 704 **8.3 ValidValueRange**

705 The ValidValueRange component is an alternative mechanism to specify the set of ValidValues  
706 for the [parent] Property component's {name}. Unlike the ValidValues component, which specifies  
707 an enumeration of values, the ValidValueRange restricts the set of valid values for a {name} by  
708 specifying a range of possible values. This mechanism can only be used when the {name} is an  
709 XML element of simpleType.

710 ValidValueRange defines an optional inclusive lower bound of the range and optional inclusive  
711 upper bound of the range. Both MAY be specified. At least one MUST be specified. The values of  
712 the lower bound and upper bound (if specified) MUST correspond to the value space definition of  
713 the {name}. If the {lowerBound} of this attribute information is NOT specified, its default value is  
714 defined by the lowest possible value defined for the value space of the {name} or "undefined".  
715 Similarly the default value of {upperBound} is the largest value for the value space of the {name}  
716 or "undefined".

717

718 If the {validValues} of a Property component is not empty and contains a ValidValueRange  
719 description, then any Web service that implements the portType or interface identified by  
720 {interface} MUST ensure that the value(s) of the resource property as identified by the {name} of  
721 the Property component MUST correspond to a value within the range specified by {validValues}.

722

723 The properties of a ValidValueRange component are as follows:

- 724
- {lowerBound} the (inclusive) lower bound of the value space defined by this component  
725 for the [parent] Property component's {name}.
  - {upperBound} the (inclusive) upper bound of the value space defined by this component  
726 for the [parent] Property component's {name}.
  - {range} a range of values, bounded by the values of {lowerBound} and {upperBound}.  
727  
728 The values within {range} MUST be compliant with any value space constraints specified  
729 on the type definition of the [parent] Property component's {name}.
- 730

731

732 The following is an XML representation of the ValidValues component:

733

```

734 <ValidValueRange
735     lowerBound="xs:anySimpleType" ? upperBound="xs:anySimpleType" ?
736     {anyAttribute}* >
737     <documentation />?
738     {any}*
739 </ValidValueRange>

```

740

741 The ValidValueRange *element information item* has the following Infoset properties:

- 742
- 743 • A [local name] of "ValidValueRange" .
  - 744 • A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".
  - 745 • one or more *attribute information items* amongst its [attributes] as follows:
    - 746 ○ one or more attribute information items amongst:
      - 747 ▪ An OPTIONAL lowerBound attribute information item
        - 748 • The value of this *attribute information item* defines an inclusive lower bound on the range of valid values to apply to the [parent] Property component's {name}.
        - 749 • The type of the lowerBound *attribute information item* is an xs:anySimpleType. This type MUST correspond to the type of the [parent] Property component's {name}.
        - 750 • The value of this *attribute information item* MUST conform to any value space constraints specified on the type definition of the [parent] Property component's {name}.
      - 751 ▪ An OPTIONAL upperBound attribute information item
        - 752 • The value of this *attribute information item* defines an inclusive upper bound on the range of valid values to apply to the [parent] Property component's {name}.
        - 753 • The type of the upperBound *attribute information item* is an xs:anySimpleType. This type MUST correspond to the type of the [parent] Property component's {name}.
        - 754 • The value of this *attribute information item* MUST conform to any value space constraints specified on the type definition of the [parent] Property component's {name}.
    - 755 ○ Zero or more namespace qualified *attribute information items*. The [namespace name] of such *attribute information items* MUST NOT be "http://docs.oasis-open.org/wsrf/rmd-1".
  - 756 • Zero or more *element information items* amongst its [children], in order as follows:
    - 757 ○ An OPTIONAL documentation *element information item* (See section 9).
- 758
- 759
- 760
- 761
- 762
- 763
- 764
- 765
- 766
- 767
- 768
- 769
- 770

- 771                   o Zero or more namespace-qualified *element information items*. The [namespace  
772 name] of such *element information items* MUST NOT be "http://docs.oasis-  
773 open.org/wsrf/rmd-1".  
774

## 775 **8.4 StaticValues**

776 The purpose of the StaticValues component is to define the minimum set of values that a [parent]  
777 Property component's {name} must contain.

778

779 If the {staticValues} of a Property component is not empty, any Web service that implements the  
780 portType or interface identified by {interface} MUST ensure that all the value(s) defined in  
781 {staticValues} appear in the {name}.

782

783 The values contained in a StaticValues component MUST conform to the XML Schema definition  
784 of the {name}. Note: because the child *element information items* of a StaticValues *element*  
785 *information item* are XML fragments, it is not required that these fragments be validated  
786 (processContents is "skip").

787

788 The properties of a StaticValues component are as follows:

- 789     • {values} zero or more XML fragments that correspond to the type of the [parent] Property  
790       component's {name}.
- 791     • The number of XML fragments within {values} MUST NOT be greater than the  
792       maxOccurs facet of the schema declaration target of the {name}.

793

794 The following is an XML representation of the StaticValues component:

795

```
796 <StaticValues  
797     {anyAttribute}* >  
798     <documentation />?  
799     {any}*  
800 </StaticValues>
```

801

802 The StaticValues *element information item* has the following Infoset properties:

- 803     • A [local name] of "StaticValues" .
- 804     • A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".
- 805     • zero or more *attribute information items* amongst its [attributes] as follows:

- 806                   o Zero or more namespace qualified *attribute information items*. The [namespace  
807                   name] of such *attribute information items* MUST NOT be "http://docs.oasis-  
808                   open.org/wsrf/rmd-1".
- 809           • Zero or more *element information items* amongst its [children], either:
- 810                   o An OPTIONAL documentation *element information item* (See section 9).
- 811                   o Zero or more namespace-qualified *element information items*. The [namespace  
812                   name] of such *element information items* MUST NOT be "http://docs.oasis-  
813                   open.org/wsrf/rmd-1".
- 814                           ▪ Each *element information item* MUST be an XML fragment that  
815                           corresponds to the type of the XML element identified by the [parent]  
816                           Property component's {name}
- 817                           ▪ Note, because these are XML fragments, it is not expected that a  
818                           processor of a MetadataDescriptor document would need to validate  
819                           these element information items (processContents = "skip").  
820

## 821 **8.5 InitialValues**

822 The purpose of the InitialValues component is to define the set of values that a [parent] Property  
823 component's {name} will contain when a WS-Resource becomes available for the first time. If the  
824 {initialValues} of a Property component is not empty, any Web service that implements the  
825 portType or interface identified by {interface} MUST ensure that all the value(s) defined in  
826 {initialValues} appear in the {name} when the service comes online. There is no guarantee as to  
827 how long these values will be present before they are modified; they are different from values  
828 defined in {staticValues} because they are mutable.

829

830 The values contained in a InitialValues component MUST conform to the XML Schema definition  
831 of the {name}. Note: because the child *element information items* of a InitialValues *element*  
832 *information item* are XML fragments, it is not required that these fragments be validated  
833 (processContents is "skip").

834

835 The properties of a InitialValues component are as follows:

- 836           • {values} zero or more XML fragments that correspond to the type of the [parent] Property  
837           component's {name}.
- 838           • The number of XML fragments within {values} MUST NOT be greater than the  
839           maxOccurs facet of the schema declaration target of the {name}.

840

841 The following is an XML representation of the InitialValues component:

842

843

844  
845  
846  
847  
848

```
<InitialValues
  {anyAttribute}* >
  <documentation />?
  {any}*
</ InitialValues>
```

849

850 The InitialValues *element information item* has the following Infoset properties:

851

- A [local name] of "InitialValues" .

852

- A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".

853

- zero or more *attribute information items* amongst its [attributes] as follows:

854

- Zero or more namespace qualified *attribute information items*. The [namespace

855

name] of such *attribute information items* MUST NOT be "http://docs.oasis-

856

open.org/wsrf/rmd-1".

857

- Zero or more *element information items* amongst its [children], either:

858

- An OPTIONAL documentation *element information item* (See section 9).

859

- Zero or more namespace-qualified *element information items*. The [namespace

860

name] of such *element information items* MUST NOT be "http://docs.oasis-

861

open.org/wsrf/rmd-1".

862

- Each *element information item* MUST be an XML fragment that

863

corresponds to the type of the XML element identified by the [parent]

864

Property component's {name}

865

- Note, because these are XML fragments, it is not expected that a

866

processor of a MetadataDescriptor document would need to validate

867

these element information items (processContents = "skip").

868

## 9 Documentation Component

869

870 The WS-Resource Metadata MetadataDescriptor specification uses the documentation *element*  
871 *information item* as a container for human readable and/or machine processable documentation  
872 in a fashion similar to that defined for WSDL 2.0 [WSDL 2.0]. The content of the *element*  
873 *information item* is "mixed" content as defined in XML Schema [XML Schema]. The  
874 documentation *element information item* may be contained by any *element information item*  
875 defined in this specification.

876

877 The following is an XML representation of the Documentation component:

878

```
879 <documentation {anyAttribute}* >  
880   {any} *  
881 </documentation>
```

882

883 The `documentation element information item` contains the following:

884

- 885 • A [local name] of "documentation".
- 886 • A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".
- 887 • Zero or more attribute information items.
  - 888 ○ Zero or more namespace qualified *attribute information items*. The [namespace  
889 name] of such *attribute information items* MUST NOT be "http://docs.oasis-  
890 open.org/wsrf/rmd-1".
- 891 • Zero or more child *element information items* amongst its [children].

891

## 10 Obtaining a MetadataDescriptor Document

892

893 There are two mechanisms that a requestor can use to obtain a WS-Resource  
894 MetadataDescriptor document:

895

- 896 1. A specific attribute extension to WSDL 1.1 portType definition
- 897 2. A specific Resource Property element.

897

### 10.1 Extending WSDL 1.1 PortType

898

899 A WS-Resource MetadataDescriptor document is associated with a WSDL 1.1 portType definition  
900 using an extension of the WSDL 1.1 portType element information item. If any aspect of the  
901 portType is associated with a MetadataDescriptor document, then the portType element MUST  
902 be extended in the manner described below. This extension is described as follows:

903

904  
905

```
906 <wsdl:definitions ...>  
907   <wsdl:portType ...  
908     wsrmd:Descriptor="xs:QName"?  
909     wsrmd:DescriptorLocation="xs:anyURI"?  
910   ... >  
911 ...  
912 </wsdl:portType>
```

913

914 This definition is further constrained as follows:

915 /wsdl:portType/@wsrmd:Descriptor

916 If this attribute appears on a WSDL 1.1 portType element its value MUST be a QName  
917 that corresponds to a MetadataDescriptor component. Further, the value of the  
918 MetadataDescriptor component contained in that document MUST have {interface} that  
919 matches the QName of the portType containing @wsrmd:Descriptor. Any service that  
920 implements this portType MUST be associated with a MetadataDescriptor that is  
921 identified by the value of this attribute.

922 /wsdl:portType/@wsrmd:DescriptorLocation

923 If this attribute appears on a WSDL 1.1 portType element its value MUST be a URI. The  
924 URI corresponds to a URL at which can be found more information about that  
925 MetadataDescriptor namespace, such as an XML document containing a Definitions  
926 element as its root element.  
927

## 928 10.2 Using Resoure Property Elements to expose 929 MetadataDescriptors

930 Clients may find and read the wsrmd:MetadataDescriptor of a WS-resource using a "metadata  
931 WS-resource" that is associated with the resource. The purpose of the metadata WS-resource is  
932 to expose a metadata document via its resource properties document. The endpoint reference of  
933 the metadata WS-resource is of type wsrmd:MetadataDescriptorReference and is exposed in the  
934 original WS-resource's resource property document; the form of this resource property is:

935

```
936 <xsd:element name="MetadataDescriptorReference"  
937           type="wsrmd:MetadataDescriptorReferenceType" />
```

938

939 The constraints on this element are as follows:

940



941 /wsrmd:MetadataDescriptorReference

942 This element is an wsa:EndpointReference to a "metadata WS-Resource" associated  
943 with the target WS-Resource. This metadata WS-Resource has a resource properties  
944 document that is equivalent in content to the metadata descriptor document of the target  
945 WS-Resource. The metadata WS-Resource MUST restrict its resource properties  
946 document such that the cardinality of the wsrmd:MetadataDescriptor child elements is  
947 one rather than zero-to-many. This allows for a single wsrmd:MetadataDescriptor to  
948 describe each WS-Resource instance.

949

950

951

## 952 11 References

### 953 11.1 Normative

- 954 [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,  
955 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 956 [URI] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI):  
957 Generic Syntax," RFC 2396, MIT/LCS, U.C. Irvine, Xerox  
958 Corporation, August 1998.
- 959 [WS-Addressing] <http://www.w3.org/2005/08/addressing.pdf>
- 960 [WS-BaseNotification] [http://docs.oasis-open.org/wsn/wsn-ws\\_base\\_notification-1.3-  
961 spec-pr-02.pdf](http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-pr-02.pdf)
- 962 [WS-Resource] [http://docs.oasis-open.org/wsr/wsr-ws\\_resource-1.2-spec-pr-  
963 02.pdf](http://docs.oasis-open.org/wsr/wsr-ws_resource-1.2-spec-pr-02.pdf)
- 964 [WS-ResourceProperties] [http://docs.oasis-open.org/wsr/wsr-ws\\_resource\\_properties-1.2-  
965 spec-pr-02.pdf](http://docs.oasis-open.org/wsr/wsr-ws_resource_properties-1.2-spec-pr-02.pdf)
- 966 [XML-Infoset] *W3C Recommendation "XML Information Set". Available at*  
967 <http://www.w3.org/TR/xml-infoset/>
- 968 [XML-Names] *W3C Recommendation "Namespaces in XML". Available at*  
969 <http://www.w3.org/TR/REC-xml-names/>  
970

### 971 11.2 Non-Normative

- 972 [WSDL2.0] W3C Recommendation "Web Services Description Language" Available at  
973 <http://www.w3.org/TR/wSDL20/>
- 974
- 975 [AppNotes] [http://www.oasis-  
976 open.org/apps/org/workgroup/wsr/download.php/16355/wsr-  
977 application\\_notes-1.2-notes-pr-02.pdf](http://www.oasis-open.org/apps/org/workgroup/wsr/download.php/16355/wsr-application_notes-1.2-notes-pr-02.pdf)  
978

## 979 **Appendix A. Acknowledgments**

980 The following individuals were members of the committee during the development of this  
981 specification:

982

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

999

1000

1001 **Appendix B. XML Schema for WS-**  
1002 **ResourceMetadataDescriptor**

1003 The XML types and elements used in this specification are defined in the following XML Schema.  
1004

```
1005 <?xml version="1.0" encoding="UTF-8"?>  
1006 <!--  
1007  
1008  
1009 OASIS takes no position regarding the validity or scope of any  
1010 intellectual property or other rights that might be claimed to pertain  
1011 to the implementation or use of the technology described in this  
1012 document or the extent to which any license under such rights might or  
1013 might not be available; neither does it represent that it has made any  
1014 effort to identify any such rights. Information on OASIS's procedures  
1015 with respect to rights in OASIS specifications can be found at the  
1016 OASIS website. Copies of claims of rights made available for  
1017 publication and any assurances of licenses to be made available, or the  
1018 result of an attempt made to obtain a general license or permission for  
1019 the use of such proprietary rights by implementors or users of this  
1020 specification, can be obtained from the OASIS Executive Director.  
1021  
1022 OASIS invites any interested party to bring to its attention any  
1023 copyrights, patents or patent applications, or other proprietary rights  
1024 which may cover technology that may be required to implement this  
1025 specification. Please address the information to the OASIS Executive  
1026 Director.  
1027  
1028 Copyright (C) OASIS Open (2005-2006). All Rights Reserved.  
1029
```

1030 This document and translations of it may be copied and furnished to  
1031 others, and derivative works that comment on or otherwise explain it or  
1032 assist in its implementation may be prepared, copied, published and  
1033 distributed, in whole or in part, without restriction of any kind,  
1034 provided that the above copyright notice and this paragraph are  
1035 included on all such copies and derivative works. However, this  
1036 document itself may not be modified in any way, such as by removing the  
1037 copyright notice or references to OASIS, except as needed for the  
1038 purpose of developing OASIS specifications, in which case the  
1039 procedures for copyrights defined in the OASIS Intellectual Property  
1040 Rights document must be followed, or as required to translate it into  
1041 languages other than English.

1042

1043 The limited permissions granted above are perpetual and will not be  
1044 revoked by OASIS or its successors or assigns.

1045

1046 This document and the information contained herein is provided on an  
1047 "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED,  
1048 INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE  
1049 INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED  
1050 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1051

1052

1053 -->

1054

1055 <xsd:schema

1056     xmlns:xsd="http://www.w3.org/2001/XMLSchema"

1057     xmlns="http://www.w3.org/2001/XMLSchema"

1058     xmlns:wsa="http://www.w3.org/2005/08/addressing"

1059     xmlns:wsrf-rp="http://docs.oasis-open.org/wsrf/rp-2"

1060     targetNamespace="http://docs.oasis-open.org/wsrf/rmd-1"

1061     xmlns:wsrmd="http://docs.oasis-open.org/wsrf/rmd-1"

1062     elementFormDefault="qualified">

1063

1064     <xsd:import

1065         namespace="http://docs.oasis-open.org/wsrf/rp-2"

1066         schemaLocation="http://docs.oasis-open.org/wsrf/rp-2.xsd" />

1067

1068     <xsd:import

1069         namespace="http://www.w3.org/2005/08/addressing"

1070         schemaLocation="http://www.w3.org/2005/08/addressing.xsd" />

1071

1072

1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115

```
<!-- ===== Utility Types ===== -->
<xsd:simpleType name="PairsOfURIType">
  <xsd:list itemType="xsd:anyURI" />
</xsd:simpleType>

<!-- ===== PortType Attribute Extensions ===== -
->
<xsd:attribute name="Descriptor" type="xsd:QName" />

<xsd:attribute name="DescriptorLocation" type="xsd:anyURI" />

<!-- ===== Documentation Component ===== -->
<xsd:complexType name="DocumentationType" mixed="true" >
  <xsd:sequence>
    <xsd:any namespace="##any"
      minOccurs="0" maxOccurs="unbounded"
      processContents="lax" />
  </xsd:sequence>
  <xsd:anyAttribute/>
</xsd:complexType>

<xsd:complexType name="DocumentedType">
  <xsd:sequence>
    <xsd:element name="documentation" type="wsrmd:DocumentationType"
      minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>

<!-- ===== Definitions Component ===== -->
<!--
<Definitions
  targetNamespace="xsd:anyURI"
  {anyAttribute}* >

  <documentation />?
  <MetadataDescriptor /> *
  {any}*

</Definitions>
-->

<xsd:complexType name="DefinitionsType" >
```

```

1116 <xsd:complexContent>
1117   <xsd:extension base="wsrmd:DocumentedType">
1118     <xsd:sequence>
1119       <xsd:element ref="wsrmd:MetadataDescriptor"
1120         minOccurs="0" maxOccurs="unbounded" />
1121       <xsd:any namespace="##other"
1122         minOccurs="0" maxOccurs="unbounded"
1123         processContents="lax" />
1124     </xsd:sequence>
1125     <xsd:attribute name="targetNamespace"
1126       type="xsd:anyURI" use="required"/>
1127     <xsd:anyAttribute namespace="##other" processContents="lax"/>
1128   </xsd:extension>
1129 </xsd:complexContent>
1130 </xsd:complexType>
1131
1132 <xsd:element name="Definitions" type="wsrmd:DefinitionsType" >
1133   <xsd:key name="MetadataDescriptor">
1134     <xsd:annotation>
1135       <xsd:documentation>
1136         To form a QName, the name of any MetadataDescriptor must be
1137         unique within a Definitions element.
1138       </xsd:documentation>
1139     </xsd:annotation>
1140     <xsd:selector xpath="wsrmd:MetadataDescriptor" />
1141     <xsd:field xpath="@name" />
1142   </xsd:key>
1143 </xsd:element>
1144
1145 <!-- ===== MetadataDescriptor Component ===== -
1146 ->
1147 <!--
1148 <MetadataDescriptor
1149   name="xsd:NCName"
1150   interface="xsd:QName"
1151   wsdlLocation="list of xsd:anyUri"?
1152   {anyAttribute}* >
1153
1154   <documentation />?
1155   <Property /> *
1156   {any}*
1157
1158 </MetadataDescriptor>

```

```

1159 -->
1160
1161 <xsd:complexType name= "MetadataDescriptorType" >
1162   <xsd:complexContent>
1163     <xsd:extension base="wsrmd:DocumentedType">
1164       <xsd:sequence>
1165         <xsd:element ref="wsrmd:Property"
1166           minOccurs="0" maxOccurs="unbounded" />
1167         <xsd:any namespace="##other"
1168           minOccurs="0" maxOccurs="unbounded"
1169           processContents="lax" />
1170       </xsd:sequence>
1171       <xsd:attribute name="name"
1172         type="xsd:NCName" use="required"/>
1173       <xsd:attribute name="interface"
1174         type="xsd:QName" use="required"/>
1175       <xsd:attribute name="wsdlLocation"
1176         type="wsrmd:PairsOfURIType" />
1177       <xsd:anyAttribute namespace="##other" processContents="lax"/>
1178     </xsd:extension>
1179   </xsd:complexContent>
1180 </xsd:complexType>
1181
1182 <xsd:element name="MetadataDescriptor"
1183   type="wsrmd:MetadataDescriptorType" />
1184
1185 <!-- ===== Property Component ===== -->
1186 <!--
1187 <Property
1188   name="xsd:QName"
1189   mutability="[constant|appendable|mutable]" ?
1190   modifiability="[read-only|read-write]" ?
1191   subscribability="xs:boolean" ?
1192   {anyAttribute}* >
1193
1194   <documentation />?
1195   [ <ValidValues> {any}* </ValidValues> |
1196     <ValidValueRange lowerBound='xsd:simpleType'
1197       upperBound='xsd:simpleType'>
1198     </ValidValueRange> ] ?
1199   <StaticValues> {any}* </StaticValues> ?
1200
1201   {any} *

```



1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244

```
</Property>
-->
<xsd:complexType name= "PropertyType" >
  <xsd:complexContent>
    <xsd:extension base="wsrmd:DocumentedType">
      <xsd:sequence>
        <xsd:choice>
          <xsd:element ref="wsrmd:ValidValues"
            minOccurs="0" maxOccurs="1" />
          <xsd:element ref="wsrmd:ValidValueRange"
            minOccurs="0" maxOccurs="1" />
        </xsd:choice>
        <xsd:element ref="wsrmd:StaticValues"
          minOccurs="0" maxOccurs="1" />
        <xsd:any namespace="##other"
          minOccurs="0" maxOccurs="unbounded"
          processContents="lax" />
      </xsd:sequence>
      <xsd:attribute name="name"
        type="xsd:QName" use="required"/>
      <xsd:attribute name="mutability"
        type="wsrmd:MutabilityType" />
      <xsd:attribute name="modifiability"
        type="wsrmd:ModifiabilityType" />
      <xsd:attribute name="subscribability" type="xsd:boolean"
        default="false" />
      <xsd:anyAttribute namespace="##other" processContents="lax"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="Property" type="wsrmd:PropertyType" />

<xsd:simpleType name="MutabilityType">
  <xsd:restriction base="xsd:string" >
    <xsd:enumeration value="constant" />
    <xsd:enumeration value="appendable" />
    <xsd:enumeration value="mutable" />
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="ModifiabilityType">
```

```

1245     <xsd:restriction base="xsd:string" >
1246         <xsd:enumeration value="read-only" />
1247         <xsd:enumeration value="read-write" />
1248     </xsd:restriction>
1249 </xsd:simpleType>
1250
1251 <!-- ===== Valid Values Component ===== -->
1252 <!--
1253 <ValidValues
1254     {anyAttribute}* >
1255     <documentation />?
1256     {any}*
1257 </ValidValues>
1258 -->
1259 <xsd:complexType name= "ValidValueType" mixed="true">
1260     <xsd:sequence>
1261         <xsd:element name="documentation" type="wsrmd:DocumentationType"
1262             minOccurs="0" maxOccurs="1" />
1263
1264         <xsd:any namespace="##other"
1265             minOccurs="0" maxOccurs="unbounded"
1266             processContents="lax" />
1267     </xsd:sequence>
1268     <xsd:anyAttribute namespace="##other" processContents="lax"/>
1269 </xsd:complexType>
1270
1271 <xsd:element name="ValidValues" type="wsrmd:ValidValueType" />
1272
1273 <!-- ===== Valid Range Component ===== -->
1274 <!--
1275 <ValidValueRange
1276     lowerBound="xs:anySimpleType" ? upperBound="xs:anySimpleType" ?
1277     {anyAttribute}* >
1278     <documentation />?
1279     {any}*
1280 </ValidValueRange>
1281 -->
1282 <xsd:complexType name= "ValidValueRangeType" mixed="true">
1283     <xsd:sequence>
1284         <xsd:element name="documentation" type="wsrmd:DocumentationType"
1285             minOccurs="0" maxOccurs="1" />
1286
1287     <xsd:any namespace="##other"

```

```

1288         minOccurs="0" maxOccurs="unbounded"
1289         processContents="lax" />
1290     </xsd:sequence>
1291     <xsd:attribute name="lowerBound" type="xsd:anySimpleType" />
1292     <xsd:attribute name="upperBound" type="xsd:anySimpleType" />
1293     <xsd:anyAttribute namespace="##other" processContents="lax"/>
1294 </xsd:complexType>
1295
1296     <xsd:element name="ValidValueRange" type="wsrmd:ValidValueRangeType"
1297 />
1298
1299 <!-- ===== Static Values Component ===== -->
1300 <!--
1301 <StaticValues
1302     {anyAttribute}* >
1303     <documentation />?
1304     {any}*
1305 </StaticValues>
1306 -->
1307 <xsd:complexType name="StaticValuesType" mixed="true">
1308     <xsd:sequence>
1309         <xsd:element name="documentation" type="wsrmd:DocumentationType"
1310             minOccurs="0" maxOccurs="1" />
1311
1312         <xsd:any namespace="##other"
1313             minOccurs="0" maxOccurs="unbounded"
1314             processContents="lax" />
1315     </xsd:sequence>
1316     <xsd:anyAttribute namespace="##other" processContents="lax"/>
1317 </xsd:complexType>
1318
1319     <xsd:element name="StaticValues" type="wsrmd:StaticValuesType" />
1320
1321 <!-- ===== Initial Values Component ===== -->
1322 <!--
1323 <InitialValues
1324     {anyAttribute}* >
1325     <documentation />?
1326     {any}*
1327 </InitialValues>
1328 -->
1329 <xsd:complexType name="InitialValuesType" mixed="true">
1330     <xsd:sequence>

```

```

1331     <xsd:element name="documentation" type="wsrmd:DocumentationType"
1332               minOccurs="0" maxOccurs="1" />
1333
1334     <xsd:any namespace="##other"
1335            minOccurs="0" maxOccurs="unbounded"
1336            processContents="lax" />
1337   </xsd:sequence>
1338   <xsd:anyAttribute namespace="##other" processContents="lax"/>
1339 </xsd:complexType>
1340
1341   <xsd:element name="InitialValues" type="wsrmd:InitialValuesType" />
1342
1343
1344 <!-- ===== MetadataDescriptorReference RP GED ===== -->
1345 <xsd:complexType name="MetadataDescriptorReferenceType">
1346   <xsd:complexContent>
1347     <xsd:extension base="wsa:EndpointReferenceType"/>
1348   </xsd:complexContent>
1349 </xsd:complexType>
1350
1351   <xsd:element name="MetadataDescriptorReference"
1352               type="wsrmd:MetadataDescriptorReferenceType" />
1353
1354 <!--
1355 Metadata Resource RP Doc
1356
1357 This defines one property - MetadataDescriptor - which must have a
1358 cardinality of one.
1359
1360 -->
1361
1362   <xsd:element name="MetadataResourceRP" type="wsrmd:DefinitionsType"/>
1363
1364 </xsd:schema>

```

## Appendix C. Revision History

Rev	Date	By Whom	What
wd-01	2004-10-07	Tom Maguire	Initial version created based on work in response to issue 10.
wd-04	2005-10-31	Dan Jemiolo	Updates to original based on TC revisions in summer/fall of 2005.
wd-06	2005-12-12	Dan Jemiolo	Clean up remaining revisions for public review. Re-inserted some features based on requests from WSDM TC.
wd-09	2006-06-04	Dan Jemiolo	Made changes related to MetadataDescriptorReference (an EPR exposed via WSRP that allows a client to read the MDD). Also (re-)added the InitialValues concept to Property.
wd-10	2006-06-19	Dan Jemiolo	Clarified nature of MetadataResourceRP and fixed some example text (T. Banks).
cd-01	2006-06-28	Dan Jemiolo	Changed status to CD.
cs-01	2006-11-13	Dan Jemiolo	Changed status to CS.

## 1368 **Appendix D. Notices**

1369 OASIS takes no position regarding the validity or scope of any intellectual property or other rights  
1370 that might be claimed to pertain to the implementation or use of the technology described in this  
1371 document or the extent to which any license under such rights might or might not be available;  
1372 neither does it represent that it has made any effort to identify any such rights. Information on  
1373 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS  
1374 website. Copies of claims of rights made available for publication and any assurances of licenses  
1375 to be made available, or the result of an attempt made to obtain a general license or permission  
1376 for the use of such proprietary rights by implementors or users of this specification, can be  
1377 obtained from the OASIS Executive Director.

1378

1379 OASIS invites any interested party to bring to its attention any copyrights, patents or patent  
1380 applications, or other proprietary rights which may cover technology that may be required to  
1381 implement this specification. Please address the information to the OASIS Executive Director.

1382

1383 Copyright (C) OASIS Open (2005). All Rights Reserved.

1384

1385 This document and translations of it may be copied and furnished to others, and derivative works  
1386 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,  
1387 published and distributed, in whole or in part, without restriction of any kind, provided that the  
1388 above copyright notice and this paragraph are included on all such copies and derivative works.  
1389 However, this document itself may not be modified in any way, such as by removing the copyright  
1390 notice or references to OASIS, except as needed for the purpose of developing OASIS  
1391 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual  
1392 Property Rights document must be followed, or as required to translate it into languages other  
1393 than English.

1394

1395 The limited permissions granted above are perpetual and will not be revoked by OASIS or its  
1396 successors or assigns.

1397

1398 This document and the information contained herein is provided on an "AS IS" basis and OASIS  
1399 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO  
1400 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE  
1401 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A  
1402 PARTICULAR PURPOSE.