



# Web Services Distributed Management: Management Using Web Services (MUWS 1.0) Part 2

Working Draft, December 2<sup>nd</sup> 2004

**Document identifier:**

wd-wsdm-muws-part2-1.0

**Location:**

<http://docs.oasis-open.org/wsdm/2004/12/wd-wsdm-muws-part2-1.0.pdf>

**Editor:**

*William Vambenepe, Hewlett-Packard* <[vbp@hp.com](mailto:vbp@hp.com)>

**Abstract:**

There are two specifications produced by the Web services Distributed Management technical committee: Management *Using* Web services (MUWS) and Management of Web services (MOWS, see [MOWS]). This document is part of MUWS.

MUWS defines how an Information Technology resource connected to a network provides manageability interfaces such that the IT resource can be managed locally or from remote locations using Web services technologies.

MUWS is composed of two parts. This document is MUWS part 2 and provides specific messaging formats used to enable the interoperability of MUWS implementations. MUWS part 1 [MUWS Part 1] provides the fundamental concepts for management using Web services. MUWS part 2 depends on MUWS part 1 while part 1 is independent of part 2.

**Status:**

This document is a working draft of version 1.0. There is no guarantee that any part of the content in this document will appear in the final, released MUWS 1.0 specification.

Committee members should send comments on this specification to the [wsdm@lists.oasis-open.org](mailto:wsdm@lists.oasis-open.org) list. Others should subscribe and send comments to the [wsdm-comment@lists.oasis-open.org](mailto:wsdm-comment@lists.oasis-open.org) list. To subscribe, send an email message to [wsdm-comment-request@lists.oasis-open.org](mailto:wsdm-comment-request@lists.oasis-open.org), with the word "subscribe" as the body of the message.

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

The errata document for this specification is maintained at:

<http://docs.oasis-open.org/wsdm/2004/12/wd-wsdm-muws-part2-1.0-errata.pdf>

## Table of Contents

38	1	Introduction.....	4
39	2	Use of the Web Services Platform.....	5
40	2.1	Use of WS-Addressing and the WS-Resource concept.....	5
41	2.2	Use of WS-Resource Properties.....	5
42	2.3	Use of WS-Notification.....	6
43	2.4	Metadata.....	6
44	2.4.1	Metadata applicable to all aspects of manageability interfaces.....	6
45	2.4.2	Metadata applicable to properties.....	7
46	2.4.3	Operations.....	8
47	2.5	Events.....	8
48	2.5.1	Event Format.....	8
49	2.5.2	Topics for capabilities.....	11
50	2.6	Representation of Categorization Taxonomies in XML.....	11
51	3	Capabilities applicable to manageable resources.....	13
52	3.1	Description.....	13
53	3.1.1	Definition.....	13
54	3.1.2	Properties.....	13
55	3.1.3	Events.....	14
56	3.2	State.....	14
57	3.2.1	Definition.....	14
58	3.2.2	Describing State Models.....	15
59	3.2.3	Information Markup Declarations.....	16
60	3.2.4	Properties.....	17
61	3.2.5	Operations.....	18
62	3.2.6	Events.....	18
63	3.3	Operational Status.....	19
64	3.3.1	Definition.....	19
65	3.3.2	Properties.....	19
66	3.3.3	Events.....	20
67	3.4	Metrics.....	20
68	3.4.1	Definition.....	20
69	3.4.2	Information Markup Declarations.....	20
70	3.4.3	Metadata.....	21
71	3.4.4	Properties.....	23
72	3.4.5	Events.....	23
73	3.5	Configuration.....	23
74	3.5.1	Definition.....	23
75	3.5.2	Properties.....	23
76	3.5.3	Operations.....	24
77	3.5.4	Events.....	24
78	4	Capabilities applicable to management in general.....	25

79	4.1	Relationships.....	25
80	4.1.1	Definition.....	25
81	4.1.2	Information Markup Declarations .....	26
82	4.1.3	Properties .....	28
83	4.1.4	Operations .....	29
84	4.1.5	Events.....	30
85	4.2	Relationship Access Capability.....	30
86	4.2.1	Definition.....	30
87	4.2.2	Events.....	31
88	4.3	Relationship Resource Capability .....	31
89	4.3.1	Definition.....	31
90	4.3.2	Properties .....	31
91	4.3.3	Events.....	32
92	4.4	Advertisement .....	32
93	4.4.1	Definition.....	32
94	4.4.2	Events.....	33
95	5	Discovery.....	35
96	5.1	Discovery using Relationships .....	35
97	5.2	Discovery using Registries.....	35
98	6	References.....	37
99	6.1	Normative.....	37
100	6.2	Non-normative.....	38
101	Appendix A.	Acknowledgements.....	39
102	Appendix B.	Notices.....	40
103	Appendix C.	Schemas.....	41
104	Appendix D.	WSDL elements .....	47
105	Appendix E.	Topics.....	49
106	Appendix F.	Description of situation types.....	51
107			

---

## 108 1 Introduction

109 This document, MUWS Part 2, builds upon the foundation provided by [MUWS Part 1]. All of the  
110 normative text presented in MUWS Part 1 is considered normative text for MUWS Part 2. All  
111 informational text presented in MUWS Part 1 is relevant informational text for MUWS Part 2.  
112 Compliance with MUWS Part 1 is REQUIRED for every aspect of MUWS Part 2.

113 The text of this specification along with Appendix C (Schemas), Appendix D (WSDL elements),  
114 Appendix E (Topics) and Appendix F (Description of situation types) is considered normative with  
115 the following exceptions: the abstract, the examples, the UML diagrams, and any section  
116 explicitly marked as non-normative.

117 The terminology and notational conventions defined in [MUWS Part 1] apply to this document.

118 The following namespaces are used, unless specified otherwise.

---

Prefix	Namespace
muws-p1-xs	<a href="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-muws-part1-1.0.xsd">http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-muws-part1-1.0.xsd</a>
muws-p2-xs	<a href="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-muws-part2-1.0.xsd">http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-muws-part2-1.0.xsd</a>
muws-p2-wsdl	<a href="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-muws-part2-1.0.wsdl">http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-muws-part2-1.0.wsdl</a>
muws-events	<a href="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-muws-events-1.0.xml">http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-muws-events-1.0.xml</a>
wsnt	<a href="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.xsd">http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.xsd</a>
wstop	<a href="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.xsd">http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.xsd</a>
wsrf-rp	<a href="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.xsd">http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.xsd</a>
wssg	<a href="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd">http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd</a>
wsdl	<a href="http://www.w3.org/2002/07/wsdl">http://www.w3.org/2002/07/wsdl</a>
wsa	<a href="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://schemas.xmlsoap.org/ws/2004/08/addressing</a>
soap	<a href="http://schemas.xmlsoap.org/soap/envelope/">http://schemas.xmlsoap.org/soap/envelope/</a>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>

---

119 XML elements ([XML 1.0 3rd Edition]) and schema ([XML Schema Part 1] and [XML Schema Part  
120 2]) types introduced in this section belong to the namespace mapped to “muws-p2-xs”.

121 WSDL ([WSDL]) elements introduced in this section belong to the namespace mapped to “muws-  
122 p2-wsdl”.

---

## 2 Use of the Web Services Platform

123

124 As a complement to the Web services platform described in [MUWS Part 1], MUWS Part 2  
125 presents an additional set of specifications in order to achieve interoperability among disparate  
126 implementations of MUWS. This goal is achieved by the precise specification of the format for  
127 each management message.

### 2.1 Use of WS-Addressing and the WS-Resource concept

128

129 MUWS Part 2 depends upon concepts presented in the Web Services Resources Framework  
130 ([WSRF]). A "manageable resource" is a refinement of a WSRF "resource". A WS-Resource, as  
131 defined by [WS-Resource], is created by composing a manageability endpoint with a manageable  
132 resource made accessible through this endpoint. In addition, a reference to a manageability  
133 endpoint relies upon reference mechanisms as defined in [WS-Resource], and more specifically,  
134 leverages and refine the endpoint reference (EPR) concept, as defined in [WS-Addressing].

135 If a manageability endpoint corresponds to zero or more manageable resources, then the  
136 "WS-Addressing Using Reference Properties Embodiment" of [WS-Resource] MUST be followed.  
137 In other words, each element listed in the *ReferenceProperties* of a WS-Resource qualified EPR  
138 MUST be included in the header of each message sent to each corresponding manageability  
139 endpoint. The MUWS specification does not currently define how to obtain an EPR. Currently, to  
140 obtain an EPR, there may be some out-of-band agreement between a service provider and a  
141 manageability consumer. Possibly, some future version of the MUWS specification might clarify  
142 and standardize an approach to obtain an EPR. This specification provides some guidelines on  
143 discovering EPRs for manageability endpoints.

144 In the specific case where a manageability endpoint corresponds to one and only one  
145 manageable resource, then either the "WS-Addressing Using Reference Properties Embodiment"  
146 concept, as above, or the "WS-Addressing Without Using Reference Properties Embodiment"  
147 concept MUST be followed. If the "WS-Addressing Without Using Reference Properties  
148 Embodiment" is followed, then the manageability endpoint does not expect to receive a list of  
149 elements in the *ReferenceProperties* of WS-Resource qualified EPR included in the message  
150 header.

151 A manageability consumer without an EPR for a manageability endpoint MAY try to invoke  
152 manageability operations without including reference properties information. If such an invocation  
153 succeeds, the manageability consumer can infer it is accessing a manageable resource through a  
154 manageability provider.

### 2.2 Use of WS-Resource Properties

155

156 Management properties as defined in MUWS are represented as WSRF "properties", and use  
157 the mechanisms defined in *WS-ResourceProperties* ([WS-RP]). In other words, each manageable  
158 resource exposes a resource properties document containing, as children of the document root,  
159 all the properties of the manageable resource. The manageable resource then makes this  
160 document available, as described in *WS-ResourceProperties*.

161 Supporting *WS-ResourceProperties* means that any implementation of an interface that includes  
162 properties MUST include access methods to these properties as defined by  
163 *WS-ResourceProperties*. Specifically, the interface MUST include the *GetResourceProperty*  
164 operation defined by [WS-RP] and MAY include the *GetMultipleProperties*,  
165 *SetResourceProperties* and *QueryResourceProperties* operations. If the  
166 *QueryResourceProperties* operation is provided, then the *QueryResourceProperties* operation  
167 SHOULD support the XPath 1.0 query expression dialect, represented by URI  
168 <http://www.w3.org/TR/1999/REC-xpath-19991116>.

## 169 2.3 Use of WS-Notification

170 MUWS uses the notification mechanism described by WS-BaseNotification ([WSN]). If a  
171 manageability capability includes an ability to offer events to a consumer, then the definition of  
172 the capability SHALL include topic space, as described in WS-Topics ([WST]). The topic space  
173 MUST contain an appropriate set of topics for the events offered by the capability. As described  
174 in MUWS Part 1, an event is defined by a “topic” QName and a “content” element. The “topic” is  
175 mapped to the topic of the event, as defined by [WST].

176 As specified by WS-BaseNotification, whether the event payload (of type *muws-p1-*  
177 *xs:ManagementEvent*) is the first child of the SOAP ([SOAP]) body or whether it is wrapped in a  
178 *wsnt:Notify* element is determined based on whether the *wsnt:UseNotify* element in the  
179 subscription message is set to *true* or *false*.

180 Note that WS-BaseNotification does not currently support a means to specify that only some of  
181 the information contained in the notification message should be sent to the consumer. MUWS  
182 does not define a means to specify this either. The manageability consumer and the implementer  
183 of a manageability endpoint should be aware that there is a performance cost for processing  
184 many, large notification messages.

## 185 2.4 Metadata

186 MUWS defines a set of base schema for metadata elements. These metadata elements can be  
187 represented as XML Schema elements. The purpose of a metadata element is to supplement the  
188 information available in the WSDL [WSDL] and the WS-ResourceProperties [WS-RP] declaration  
189 for a manageability interface. A metadata element provides additional description relevant to the  
190 managed resource. In particular, a metadata element enables a tool or management application,  
191 to perform detailed reasoning and make specialized inferences about a manageable resource at  
192 runtime, and, during development, when no instance is available for a manageable resource.

193 If metadata is required, then an XML document containing metadata is defined and associated  
194 with a WS-ResourceProperties document and WSDL. Document processing, like an XPath  
195 query, is used to extract all or part of the metadata. Currently, WSDM does not define the format  
196 of, how to associate, or, how to access document metadata content. Although some mechanism  
197 is necessary, this MUWS specification does not provide any mechanism for accessing metadata  
198 from an instance of a manageable resource.

199 Also, this MUWS specification does not provide any description of how metadata is associated  
200 with a type of manageable resource, is stored, or made available.

201 The MUWS specification defines a set of metadata elements that apply to the basic  
202 manageability of a manageable resource. The MUWS specification uses Global Element  
203 Declarations to represent a metadata element.

### 204 2.4.1 Metadata applicable to all aspects of manageability interfaces

205 MUWS defines metadata elements applicable to all aspects of a manageability interface  
206 (operations, properties, events...). These elements are:

207

208 `<muws-p2-xs:Capability>xs:anyURI</muws-p2-xs:Capability> *`

209 **muws-p2-xs:Capability** metadata element SHOULD be provided for any MUWS aspect of a  
210 manageability interface. This enables discovery of aspects of an interface associated with a  
211 capability. This element contains a URI identifying the capability.

212 This metadata element indicates the classification of an aspect of an interface according to an  
213 intended capability, or capabilities. For example, an aspect may be classified as a metric, or, as  
214 a configuration property. A property may be relevant to more than one capability. For example, a

215 configuration property of a computer system contains the IP address but this same property could  
216 also be used for identification purposes.

217 Some of the known capabilities are listed below for illustration. This is not an exhaustive list. For a  
218 detailed explanation, see the relevant MUWS manageability capability specification. Additional  
219 capabilities are expected to be added as extensions to MUWS.

- 220 • <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Identity>  
221 Identity capability. See [MUWS Part 1].
- 222 • <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Configuration>  
223 Configuration property. See section 3.5.
- 224 • <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/CorrelatableProperties>  
225 "Correlatable Properties" capability. See [MUWS Part 1].
- 226 • <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/State>  
227 State capability. See section 3.1.3.
- 228 • <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Metrics>  
229 Metrics capability. See section 3.4.
- 230 • *User defined*  
231 A user defined capability that extends, or, is different from, a standard capability defined  
232 in MUWS.

233  
234

```
235 <muws-p2-xs:ValidWhile Dialect="xs:anyURI" > {any} * </muws-p2-  
236 xs:ValidWhile>
```

237 **muws-p2-xs:ValidWhile** contains a statement that, when true, asserts that the interface aspect  
238 to which this metadata element is related is valid. This is used, for example, to express the fact  
239 that an operation can only be invoked when certain properties have certain values.

240 **muws-p2-xs:ValidWhile/@muws-p2-xs:Dialect** is a URI identifying how the statement in *muws-*  
241 *p2-xs:ValidWhile* is built and what rules govern its evaluation. MUWS defines one possible value  
242 for this element. Other values can also be defined.

243 The value defined by MUWS is <http://www.w3.org/TR/1999/REC-xpath-19991116>. When this  
244 dialect is used, the content of *muws-p2-xs:ValidWhile* is an [XPath 1.0] expression. This  
245 expression is evaluated against the resource properties document of the manageable resource. If  
246 the XPath expression evaluates to a Boolean value of *true*, or if it evaluates to a non-empty non-  
247 boolean value without any errors, then the statement is considered true.

## 248 2.4.2 Metadata applicable to properties

249 General purpose metadata that is not management specific is defined in the MUWS specification,  
250 but not specified in schema. General purpose metadata that can be defined for any property  
251 include:

- 252 • *Mutability* – indicates if the property value can change over time
- 253 • *Modifiability* – indicates if the property can be set directly (not as a side-effect)
- 254 • *Valid Values* – a set of valid values for the property
- 255 • *Valid Range* – a range of valid values for the property
- 256 • *Static Values* – a set of permanent values for the property
- 257 • *Notifiability* – indicates if a notification is sent when there is a change to the value of the  
258 property

259 Schema to represent general purpose metadata should be composed from a metadata  
260 specification, for example, the WS-Resource Metadata Descriptor [WSRMD], as developed in the  
261 WS-RF OASIS technical committee.

262 In addition, MUWS defines a set of metadata related to management. Any property element may  
263 have the following manageability metadata element:



264 `<muws-p2-xs:Units>xs:string</muws-p2-xs:Units>`

265 **muws-p2-xs:Units** indicates the default unit for this property as a string.

266 Other metadata elements, applicable for metric-type properties, are defined in section 3.4.3.

## 267 2.4.3 Operations

268 General purpose metadata, that is not management specific, is defined in the MUWS  
269 specification, but not specified in schema. General purpose metadata that can be defined for any  
270 operation includes:

- 271 • *Idempotency* – indicates if invoking the operation twice is equivalent to invoking it once

272 Schema to represent general purpose metadata should be composed from a metadata  
273 specification, for example, the WS-Resource Metadata Descriptor [WSRMD], as developed in the  
274 WS-RF OASIS technical committee.

275 In addition, MUWS defines metadata related to management. Any operation element may have  
276 the following manageability metadata element:

277

```
278 <muws-p2-xs:PostCondition Dialect="xs:anyURI">  
279   {any} *  
280 </muws-p2-xs:PostCondition>
```

281 **muws-p2-xs:PostCondition** contains a statement that asserts "true" immediately after the  
282 corresponding operation is complete.

283 **muws-p2-xs:PostCondition/@muws-p2-xs:Dialect** is a URI identifying how the statement in  
284 *muws-p2-xs:PostCondition* is built, and what rules govern its evaluation. MUWS defines one  
285 possible value for this element. Other values can be defined.

286 The value defined by MUWS is <http://www.w3.org/TR/1999/REC-xpath-19991116>. When this  
287 dialect is used, the content of *muws-p2-xs:PostCondition* is an [XPath 1.0] expression. This  
288 expression is evaluated against the resource properties document of the manageable resource. If  
289 the XPath expression evaluates to a Boolean value of *true*, or, if it evaluates to a non-empty non-  
290 boolean value without any errors, then the statement is considered true.

## 291 2.5 Events

### 292 2.5.1 Event Format

293 [MUWS Part 1] defines the *muws-p1-xs:ManagementEvent* Global Element Declaration as a  
294 container for management events. *muws-p1-xs:ManagementEvent* allows information to be  
295 added via extensibility elements. The *muws-p2-xs:Situation* element defined below MUST be  
296 present as a child of the *muws-p1-xs:ManagementEvent* element in notifications.

297 As a result, the event format is flexible and extensible. At the same time, automated analysis is  
298 possible, as the event format provides a means to classify an event into one of a limited set of  
299 classifications and sub-classifications.

300 MUWS event classifications are based on a thorough analysis of event types, as produced by a  
301 wide range of IT equipment, and grouped according to the general nature of events. For example,  
302 virtually all manageable resources have a means of being started. However, almost all managed  
303 resources express a start event in some unique way. The basic knowledge that the resource has  
304 started is all that is necessary, even for fairly sophisticated, automated management.

305 To support event classifications, the MUWS specification defines the *SituationCategoryType*  
306 element, a specialization of a *muws-p2-xs:CategoryType*. MUWS defines the top level of  
307 classifications. Extensions to these classifications enable a refined event classification. Through



308 the use of the extensible *muws-p2-xs:CategoryType* mechanism, WSDM event consumers can  
309 comprehend the situation for an event to a degree commensurate with their ability.

```
310 <muws-p2-xs:Situation>
311   <muws-p2-xs:SituationCategory>
312     muws-p2-xs:SituationCategoryType
313   </muws-p2-xs:SituationCategory>
314   <muws-p2-xs:SuccessDisposition>
315     (Successful|Unsuccessful)
316   </muws-p2-xs:SuccessDisposition> ?
317   <muws-p2-xs:SituationTime>xs:dateTime</muws-p2-xs:SituationTime> ?
318   <muws-p2-xs:Priority>xs:short</muws-p2-xs:Priority> ?
319   <muws-p2-xs:Severity>xs:short</muws-p2-xs:Severity> ?
320   <muws-p2-xs:Message>muws:LangString</muws-p2-xs:Message> ?
321   <muws-p2-xs:SubstitutableMsg MsgId="xs:string" MsgIdType="xs:anyURI">
322     <muws-p2-xs:Value>xs:anySimpleType</muws-p2-xs:Value>*
323   </muws-p2-xs:SubstitutableMsg> ?
324 </muws-p2-xs:Situation>
```

325 **muws-p2-xs:Situation/muws-p2-xs:SituationCategory** categorizes the type of the situation  
326 that caused the event report. The values, listed below, represent the names of elements in the  
327 *muws-p2-xs* namespace. The categories are listed in the order of precedence. In a case where  
328 there may be some ambiguity about which category to use, the higher precedent category  
329 SHOULD be used. The ordering of situation categories is based on experimental data showing  
330 relative importance of various types of events. The use of a higher precedent category permits  
331 more effective and timely correlation and analysis of events that may indicate the presence of a  
332 serious problem. Details and examples for use of the following values are documented in  
333 Appendix F. This element is REQUIRED.

- 334 • AvailabilitySituation
  - 335 • CapabilitySituation
  - 336 • ConfigureSituation
  - 337 • StopSituation
  - 338 • StartSituation
  - 339 • RequestSituation
  - 340 • DestroySituation
  - 341 • CreateSituation
  - 342 • DependencySituation
  - 343 • ConnectSituation
  - 344 • ReportSituation
  - 345 • OtherSituation
- 346

347 **muws-p2-xs:Situation/muws-p2-xs:SuccessDisposition** in the case where this situation is  
348 triggered by a command, this value specifies a successful disposition of the command causing a  
349 report of this situation. This element is OPTIONAL and should not be included if the situation is  
350 not the result of a command. The element is a restriction of the type *xs:string* allowing the  
351 following values:

- 352 • Successful
  - 353 • Unsuccessful
- 354

355 **muws-p2-xs:Situation/muws-p2-xs:SituationTime** represents the date and time an event is  
356 observed. If the value does not include a time zone designation, or, if the value does not use 'Z'  
357 for UCT, then the value MUST be interpreted as having a time zone of UCT. The value of  
358 SituationTime MUST provide granularity as precise as supported by the generating platform. This  
359 is a REQUIRED element and MUST be provided by the component acting as the originator of an  
360 event.

361 **muws-p2-xs:Situation/muws-p2-xs:Priority** represents the importance of an event. This  
362 element supports management functions requiring an event to be associated with a priority. This  
363 is an OPTIONAL element. Values are constrained to a range from 0 through 100. The predefined  
364 priorities are:

- 365 • Low (10)
- 366 • Medium (50)
- 367 • High (70).

368 Other priorities MAY be used but MUST NOT be less than 0 or greater than 100.

369 **muws-p2-xs:Situation/muws-p2-xs:Severity** represents the perceived severity of the status the  
370 event is describing with respect to the application that reports the event. This element supports  
371 management functions requiring an event to be associated with a severity. This is an OPTIONAL  
372 element. Severity levels, based upon the DMTF CIM Alert Indications Perceived Severity, are as  
373 follows:

- 374 • 6 (Fatal): a condition is unrecoverable and the service is no longer available.
- 375 • 5 (Critical): a condition affecting the service has occurred. Immediate corrective action is  
376 required.
- 377 • 4 (Major): a problem of relatively high severity has occurred. It is likely that normal use of  
378 the service is impeded.
- 379 • 3 (Minor): a problem of relatively low severity has occurred. It is unlikely that normal use  
380 of the service is impeded.
- 381 • 2 (Warning): a problem affecting the service may occur. Diagnostic and corrective action  
382 is recommended.
- 383 • 1 (Information): a message output considered as normal and expected. For example, a  
384 process begins, a process finishes, or status information is displayed.
- 385 • 0 (Unknown): a severity level cannot be determined.

386

387 **muws-p2-xs:Situation/muws-p2-xs:Message** represents the text accompanying an event. This  
388 is typically the resolved message string in a human-readable format, as rendered for a specific  
389 locale, and is of type *muws-p2-xs:LangString* which is an extension of *xs:string* requiring the  
390 *xml:lang* attribute. This is an OPTIONAL property. While the string length for *Message* is  
391 unbounded, it is RECOMMENDED that the string length for *Message* does not exceed 1024  
392 characters.

393 **muws-p2-xs:Situation/muws-p2-xs:SubstitutableMsg** – represents the message data in a  
394 substitutable form. The attributes *MsgId* and *MsgIdType* identify the base message type and  
395 text. The element value contains the data that will be formatted according to the formatting rules  
396 defined by the *MsgId*. This is an OPTIONAL element. However, if this element is used, it must  
397 contain all the attributes and elements specified below.

398 **muws-p2-xs:Situation/muws-p2-xs:SubstitutableMsg/@muws-p2-xs:MsgId** specifies the  
399 message identifier of an event. This identifier SHOULD be a unique value string, consisting of  
400 alphanumeric or numeric characters. The value can be as simple as a string of numeric  
401 characters that identify a message in a message catalog. As an alternative, the value can be a  
402 multipart string of alphanumeric characters, for example, DBT1234E. This is a REQUIRED  
403 attribute. The maximum string length for *MsgId* MUST NOT exceed 256 characters. The  
404 *MsgIdType* attribute indicates the formatting type of the *MsgId*.

405 **muws-p2-xs:Situation/muws-p2-xs:SubstitutableMsg/@muws-p2-xs:MsgIdType** specifies  
406 the meaning and format of the *MsgId*. This is a REQUIRED attribute. The type of the *MsgIdType*  
407 attribute is a URI.

408 **muws-p2-xs:Situation/muws-p2-xs:SubstitutableMsg/muws-p2-xs:Value** can be of any  
409 simple type. There are one or more occurrences of this element with each occurrence containing  
410 an *xsi:type* attribute defining the type of the contained data. This element is used to pass data  
411 values that are substituted as a message is formatted. This element is OPTIONAL. A *MsgId* and

412 *MsgIdType* define rules to map parameters into a composed message, based upon the order of  
413 the *Value* elements.

414 As an example, a minimal *SituationType* report for the initiation of a requested restart (at 6:06PM  
415 in Greenwich on Nov 11, 2004) would be as follows.

```
416 <muws-p2-xs:Situation>  
417   <muws-p2-xs:SituationCategory>  
418     <foo:RestartInitiated>  
419       <muws-p2-xs:StartSituation/>  
420     </foo:RestartInitiated>  
421   </muws-p2-xs:SituationCategory>  
422   <muws-p2-xs:SuccessDisposition>Successful</muws-p2-xs:SuccessDisposition>  
423   <muws-p2-xs:SituationTime>2004-11-11T18:06:00Z  
424   </muws-p2-xs:SituationTime>  
425   <muws-p2-xs:Message xml:lang="en">  
426     Managed Thing XXX: restart processing begun  
427   </muws-p2-xs:Message>  
428 </muws-p2-xs:Situation>
```

429 Please note, as outlined in the description of *muws-p2-xs:CategoryType*, the most general  
430 situation classification appears as the innermost element within the XML nest.

## 431 2.5.2 Topics for capabilities

432 For each capability defined by MUWS, topics are defined that encompasses every event related  
433 to that capability. For example, if a property related to capability "foo" changes, then a notification  
434 is sent to subscribers of the topic corresponding to a change event on this property, as described  
435 by [WS-RP]. Concurrently, since this property is associated with the "foo" capability, a notification  
436 is also sent to subscribers of the topic encompassing change events associated with capability  
437 "foo".

438 Appendix E contains the XML description of all the topics defined in the MUWS specification. The  
439 sections of this document that define a capability also define the topic(s) associated with that  
440 capability. The following MUWS topics encompass every event associated with the capability  
441 defined in MUWS Part 1:

442 The *muws-events:IdentityCapability* topic defined below is used for events related to the *Identity*  
443 capability.

```
444 <wstop:Topic name="IdentityCapability"  
445   messageTypes="muws-pl-xs:ManagementEvent ">  
446 </wstop:Topic>
```

447 The *muws-events:ManageabilityCharacteristicsCapability* topic defined below is used for events  
448 related to the *ManageabilityCharacteristics* capability.

```
449 <wstop:Topic name="ManageabilityCharacteristicsCapability"  
450   messageTypes="muws-pl-xs:ManagementEvent ">  
451 </wstop:Topic>
```

452 The *muws-events:CorrelatablePropertiesCapability* topic defined below is used for events related  
453 to the *CorrelatableProperties* capability.

```
454 <wstop:Topic name="CorrelatablePropertiesCapability"  
455   messageTypes="muws-pl-xs:ManagementEvent ">  
456 </wstop:Topic>
```

## 457 2.6 Representation of Categorization Taxonomies in XML

458 In the description of several manageability capabilities, categories of information are organized in  
459 taxonomies. This is for example the case for the categories of relationships between manageable

460 resources, for operational states of resources, etc. In order to convey category information,  
461 including taxonomy lineage, to a manageability consumer, and, in order to represent XML  
462 information instances, the following convention is used:

463 MUWS defines an XML Schema complex type called `CategoryType`. The content of XML  
464 elements of this type is any XML element. When an element is defined of this type, it MUST obey  
465 the following rules:

- 466 • The element and each descendant has, at most, one child element.
- 467 • The top-level element and each descendant represent one category in a taxonomy.
- 468 • The top level element represents the most specialized category. Each element  
469 represents a more specialized category than the category represented by the element it  
470 contains, if any.

471 The `CategoryType` XML Schema type is declared as follows:

```
472 <xs:complexType name="CategoryType">  
473 <xs:sequence>  
474 <xs:any namespace="##any" minOccurs="0" processContents="lax" />  
475 </xs:sequence>  
476 </xs:complexType>
```

477 The `CategoryType` type is used to declare an XML element containing instances of general, or  
478 unqualified, category information. The `CategoryType` type is also used to derive an XML Schema  
479 type representing a specific category, for example, a relationship among resources, or among  
480 operational states.

481 Category information MUST be declared as follows:

- 482 • An XML element declaring which QName identifies the semantics of the category.
- 483 • The XML element declaring an XML Schema type which is a restriction of `muws-p2-`  
484 `xs:Category`, or a specialized XML Schema type derived from some other refinement of  
485 `muws-p2-xs:Category`, for example, `muws-p2-xs:RelationshipType`.
- 486 • The contents of the XML element MUST be either:
  - 487 • The one XML element corresponding to the generalization of the currently declared  
488 category
  - 489 • The empty sequence. This case occurs if the declared category does not have any  
490 generalizations. For example, the declared category might be the top of a taxonomy.

491 For example, assume that information about a maintenance state is represented, using the  
492 approach described above. In this example, "off-for-maintenance" is a substate of "offline", which  
493 is a substate of a resource being "unavailable". The XML representation for this example follows:

```
494 <mydomain:Off-for-Maintenance>  
495 <mydomain:Offline>  
496 <anyresource:Unavailable/>  
497 </mydomain:Offline>  
498 </mydomain:Off-for-Maintenance>
```

499 By processing the XML information, a manageability consumer may learn that a resource is in a  
500 state identified by the `mydomain:Off-for-Maintenance` element. However, at the same time, if the  
501 manageability consumer is not aware of definitions and semantics associated with the `mydomain`  
502 namespace, the consumer may safely assume the resource is in the commonly known state  
503 identified by `anyresource:Unavailable`. Since the most specialized elements are first encountered,  
504 a consumer can generally stop processing an element of type `muws-p2-xs:Category` as soon as it  
505 reaches an element the semantic of which it understands.

---

## 506 3 Capabilities applicable to manageable 507 resources

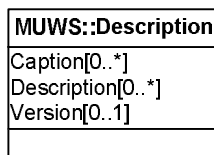
508 This section defines capabilities applicable to manageable resources. The capabilities defined in  
509 this section complement the capabilities defined in MUWS Part 1.

### 510 3.1 Description

511 The manageability capability URI for the description capability is  
512 <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Description>

#### 513 3.1.1 Definition

514 Figure 1 shows a UML representation of the *Description* capability.



515  
516

Figure 1: MUWS Description

#### 517 3.1.2 Properties

518 This capability defines the following properties:

```
519 <muws-p2-xs:Caption>muws-p2-xs:LangString</muws-p2-xs:Caption> *
```

520 **muws-p2-xs:Caption** contains a descriptive name for the manageable resource.. The *Caption*  
521 property is intended for human consumption. A *Caption* is expected to be short and is suitable for  
522 display next to a graphic icon. *Caption* is a read-write, optional property with a cardinality of 0 to  
523 many. *Caption* is of type *muws-p2-xs:LangType*, which is a restriction of *xs:string* carrying an  
524 *xml:lang* attribute. This attribute contains a language identifier as defined by [RFC3066]. There  
525 can not be more than one *Caption* per language identifier.

526 Metadata for *Caption*:

527 It is *Mutable*

528 It is *Modifiable*

529 It has the following *Capability* metadata item:

```
530 <muws-p2-xs:Capability>  
531   http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Description  
532 </muws-p2-xs:Capability>
```

533

```
534 <muws-p2-xs:Description>muws-p2-xs:LangString</muws-p2-xs:Description> *
```

535 **muws-p2-xs:Description** is a string containing a description for the resource being managed.  
536 The *Description* property is intended for human consumption. A *Description* is expected to be  
537 longer and more detailed than a *Caption*. *Description* is a read-write optional property with a  
538 cardinality of 0 to many. *Description* is of type *muws-p2-xs:LangType*, which is a restriction of  
539 *xs:string* carrying an *xml:lang* attribute. This attribute contains a language identifier as defined by  
540 [RFC3066]. There cannot be more than one *Description* per language identifier.

541 Metadata for *Description*:

542 It is *Mutable*  
543 It is *Modifiable*  
544 It has the following *Capability* metadata item:

```
545 <muws-p2-xs:Capability>  
546   http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Description  
547 </muws-p2-xs:Capability>
```

548

```
549 <muws-p2-xs:Version>xs:string</muws-p2-xs:Version> ?
```

550 **muws-p2-xs:Version** is a string representing the version of the resource being managed. MUWS  
551 does not specify how this string is constructed. The *Version* string can be specified by any  
552 domain-specific specification that uses MUWS. *Version* is an optional property with a cardinality  
553 of 0 to1.

554 Metadata for *Version*:

555 It is *Mutable*  
556 It is *Modifiable*  
557 It has the following *Capability* metadata item:

```
558 <muws-p2-xs:Capability>  
559   http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Description  
560 </muws-p2-xs:Capability>
```

### 561 3.1.3 Events

562 The *muws-events:DescriptionCapability* topic defined below is used for events related to the  
563 *Description* capability.

```
564 <wstop:Topic name="DescriptionCapability"  
565   messageTypes="muws-p1-xs:ManagementEvent" >  
566 </wstop:Topic>
```

## 567 3.2 State

568 The manageability capability URI for the State capability is  
569 <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/State>

### 570 3.2.1 Definition

571 A resource may exhibit behavior according to one or more state models. Since a single definition  
572 of an operational state model is not sufficient for all types of resource, the *State* capability is a  
573 means to allow different state models to be used by different resources. The state capability  
574 provides a pattern for representing any type of state or state model that a manageable resource  
575 can expose. This section uses operational state as an example to illustrate the application of this  
576 pattern to a simple state model.

577 Although MUWS defines no state model, there should be a very limited and well defined set of  
578 states to facilitate interoperability. Each state is identified by a URI. This URI is exposed by a  
579 resource via some resource property.

580 This capability does not define any specific property, operation or event. A manageability  
581 endpoint is said to provide this capability if at least one property exposes state information and  
582 follows the pattern described in section 3.2.3.2.

583 **3.2.2 Describing State Models**

584 Each state in a state-machine has a well-defined meaning. It is possible to reuse state definitions  
585 in different state machines. States are identified by an element with a particular QName, using  
586 the taxonomy scheme defined in section 2.6.

587 States in the state model may have duration. Transitions between states are considered to be  
588 instantaneous.

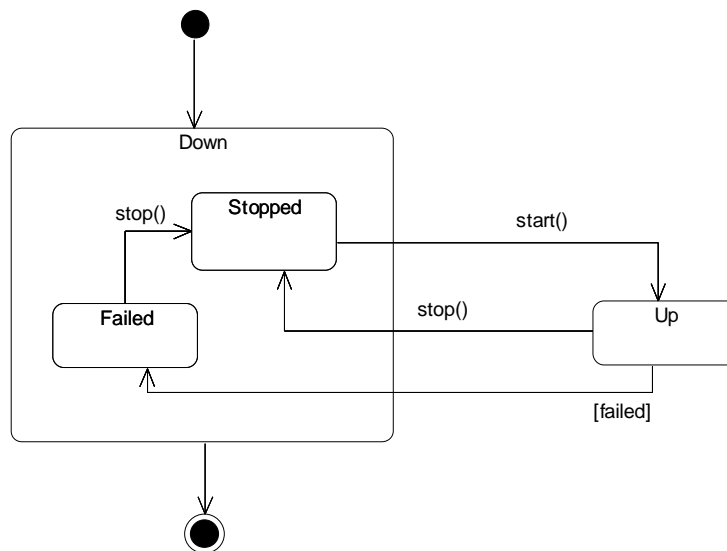
589 States can have sub-states that MUST be wholly contained within a higher-level state.

590 A state model may also define an operation that can be used to affect some transition in the  
591 model. Note that a transition may also occur as a result of some internal or external event on the  
592 resource.

593 Each state machine has an associated resource property element exposing a read-only view of  
594 the current state of the state machine. Therefore, a consumer cannot change a resource state by  
595 modifying a state resource property.

596 There may be more than one possible transition between two states in the state model. The  
597 individual transitions between states are identified by a URI. This identification allows, for  
598 example, a receiver of state transition notifications to discern which transition occurred.

599 Figure 2 shows a simple state model that is used as an example in this section – it does not  
600 constitute the specification of a recommended state model.



601

602

Figure 2: Example Operational State Model

603 In this example, the state machine is identified by URI

604 <http://example.com/StateModels/SimpleOperationalState>, bound to namespace prefix *exns*.

605 In this example, the state model has four states. Each state is represented by elements with a  
606 QName, as follows:

- 607 • *exns:Down*

608 This QName corresponds to the “Down” state in the UML diagram. A resource in this  
609 state is unable to perform any of its functional tasks.

- 610 • *exns:Stopped*

611 This QName corresponds to the “Stopped” sub-state of the “Down” state in the UML  
612 diagram. Since this state is a sub-state of the “Down” state, it follows that a resource in



- 613 the "Stopped" sub-state is unable to perform any of its functional tasks. A manageable  
 614 resource exposing this state model can be started from the "Stopped" sub-state.
- 615 • *exns:Failed*
- 616 This QName corresponds to the "Failed" sub-state of the Down state in the UML diagram.  
 617 Since this state is a sub-state of the "Down" state, it follows that a resource in the "Failed"  
 618 sub-state is unable to perform any of its functional tasks. A manageable resource  
 619 exposing this state model can not be started directly from the "Failed" sub-state. Such a  
 620 resource must first transition to the "Stopped" sub-state.
- 621 • *exns:Up*
- 622 This QName corresponds to the "Up" state in the UML diagram. A resource in this state is  
 623 able to perform at least some of its functional tasks.

## 624 3.2.3 Information Markup Declarations

### 625 3.2.3.1 Representation of States

626 A state, as represented in a state model, may be a top level state or a state that is nested within  
 627 another state according to some defined taxonomy. MUWS defines a way to represent a state  
 628 category and its taxonomy lineage, but an actual definition of any category is specific to a  
 629 particular resource management model. Therefore MUWS defines no state model. In other  
 630 words, MUWS specifies only the mechanism used to convey a state category in XML. The  
 631 MUWS mechanism applied to the representation of states is defined as follows:

632 *muws-p2-xs:StateType* XML Schema type is declared as follows

```
633 <xs:complexType name="StateType">
634   <xs:complexContent>
635     <xs:extension base="muws-p2-xs:CategoryType" />
636   </xs:complexContent>
637 </xs:complexType>
```

638 The *muws-p2-xs:StateType* type is used to declare an XML element containing an instance of  
 639 state.

640 A state MUST be declared as follows:

- 641 • An XML element declaring which QName identifies the semantics of the state.
- 642 • The XML element has an XML Schema type of *muws-p2-xs:StateType*, or a restriction of  
 643 *muws-p2-xs:StateType*.
- 644 • The contents of the XML element MUST be either:
  - 645 • The one XML element that corresponds to the state containing this state. In other  
 646 words, this state is a sub-state of another state.
  - 647 • The empty sequence. This case occurs if this state is not a sub-state of another  
 648 state.

649 For example, the "Failed" state in the example above is a sub-state of the "Down" state. An  
 650 instance of the "Failed" state may be represented, using the rules described above, by the  
 651 following XML fragment:

```
652 <my:StateTypeInstanceElement xsi:type="StateType">
653   <exns:Failed>
654     <exns:Down/>
655   </exns:Failed>
656 </my:StateTypeInstanceElement>
```

### 657 3.2.3.2 Representation of state

658 MUWS defines the following Global Element Declaration (GED) to represent an instance of a  
659 state:

```
660 <muws-p2-xs:State>muws-p2-xs:StateType</muws-p2-xs:State>
```

661 The State element provides a representation of the state of a manageable resource. The State  
662 element follows the convention for the *muws-p2-xs:CategoryType* type described in section 2.6.  
663 This convention allows the rendering of a hierarchy of states and sub-states. State values are  
664 defined in the operational state model for the resource. This specification does not define the  
665 operational state model for any resource.

### 666 3.2.3.3 Representation of state transition

667 MUWS defines the following Global Element Declaration (GED) which contains an XML  
668 representation of a change of state in a state model.

```
669 <muws-p2-xs:StateTransition Time"xs:dateTime"  
670 TransitionIdentifier=" xs:anyURI" ?>  
671 <muws-p2-xs:EnteredState>muws-p2-xs:StateType</muws-p2-xs:EnteredState>  
672 <muws-p2-xs:PreviousState>muws-p2-xs:StateType</muws-p2-  
673 xs:PreviousState>?  
674 {any} *  
675 </muws-p2-xs:StateTransition>
```

676 **muws-p2-xs:StateTransition** is used for representing information about a state change.

677 **muws-p2-xs:StateTransition/@muws-p2-xs:Time** attribute indicates the time at which the  
678 transition occurred (transitions are assumed to be instantaneous). This attribute is REQUIRED.

679 **muws-p2-xs:StateTransition/@muws-p2-xs:TransitionIdentifier** attribute indicates the actual  
680 transition that occurred. This attribute is OPTIONAL and may be omitted where, for example,  
681 there is only one transition between the *EnteredState* and the *PreviousState*.

682 **muws-p2-xs:StateTransition/muws-p2-xs:EnteredState** element indicates which state has  
683 been entered during the transition. This element is REQUIRED.

684 **muws-p2-xs:StateTransition/muws-p2-xs:PreviousState** element indicates the state that the  
685 resource was in immediately prior to the state change occurring. This element is OPTIONAL to  
686 allow for the time between the state model being created in some initial state, for example when  
687 the resource is created, and the time of the transition from that initial state.

### 688 3.2.4 Properties

689 This capability does not define any standard property.

690 A capability defining a state model SHOULD define a resource property that exposes the state., It  
691 is RECOMMENDED that a state model also define a resource property that exposes the last  
692 state transition.

693 The property used to expose the state must either contain the *muws-p2-xs:State* element or be of  
694 type *muws-p2-xs:StateType*. The name of the property can be any name meaningful to the state  
695 model defined in the capability. There may be multiple state capabilities, and therefore multiple  
696 state properties for a resource. The metadata for this property SHOULD include the possible  
697 values. That is, the state model should provide a list of states in the state model.

698 The property to represent the last transition, if such a property is provided, must contain the  
699 element *muws-p2-xs:StateTransition*. The name of the last transition property can be any name  
700 meaningful to the state model. There may be multiple state capabilities and multiple properties  
701 exposing the last transition.

### 702 3.2.4.1 Example

703 Examples of resource properties for an operational state capability could be specified as follows:

```
704 <foo:OperationalState>
705   <muws-p2-xs:State>...</muws-p2-xs:State>
706 </foo:OperationalState>
707 <foo>LastOperationalStateTransition>
708   <muws-p2-xs:StateTransition>...</muws-p2-xs:StateTransition>
709 </foo>LastOperationalStateTransition>
```

710 The following fragment provides an example from a resource properties instance document  
711 containing the properties defined in this example:

```
712 <foo:OperationalState>
713   <muws-p2-xs:State>
714     <exns:Failed><exns:Down/></exns:Failed>
715   </muws-p2-xs:State>
716 </foo:OperationalState>
717 <foo>LastOperationalStateTransition>
718   <muws-p2-xs:StateTransition Time="2004-03-11T11:30:56Z"
719   TransitionIdentifier="http://example.com/SimpleOperationalState/T/Failed">
720     <muws-p2-xs:EnteredState>
721       <exns:Failed><exns:Down/></exns:Failed>
722     </muws-p2-xs:EnteredState>
723     <muws-p2-xs:PreviousState>
724       <exns:Up/>
725     </muws-p2-xs:PreviousState>
726   </muws-p2-xs:StateTransition>
727 </foo>LastOperationalStateTransition>
```

728 In this example, the *foo:OperationalState* property contains the current operational state of the  
729 resource, using the *muws-p2-xs:State* element defined in section 3.2.3.2. The  
730 *foo>LastOperationalStateTransition* property contains a description of the most recent operational  
731 state transition for the resource, using the *muws-p2-xs:StateTransition* element as defined in  
732 section 3.2.3.2.

### 733 3.2.5 Operations

734 A capability defining a state model usually defines any operations that can be used to cause  
735 some of the transitions within the state model. These operations are specific to the resource and  
736 its state model.

### 737 3.2.6 Events

738 The *muws-events:StateCapability* topic defined below is used for events related to the *State*  
739 capability.

```
740 <wstop:Topic name="StateCapability"
741   messageTypes="muws-pl-xs:ManagementEvent">
742 </wstop:Topic>
```

743 It is RECOMMENDED that resources send a notification on a transition between states. The topic  
744 defined for the *State* capability SHALL be used to publish such notifications. If a resource sends  
745 such a notification, then the notification message MUST contain at least the XML element  
746 representing a state transition (*muws-p2-xs:StateTransition*).

747 To obtain events about a certain state transition, a subscriber can use a *Selector*, on the  
748 notification subscription, to select only those events containing the required *muws-p2-*  
749 *xs:TransitionIdentifier* element in the notification content, or, a combination of *muws-p2-*  
750 *xs:EnteredState* and *muws-p2-xs:PreviousState* elements in the notification content. The *Selector*  
751 mechanism is described in [WSN].

752 To filter for events about entry into a particular state or set of states, a Selector expression based  
753 on the *muws-p2-xs:EnteredState* element can be used. To filter for events about exit from a  
754 particular state or set of states a *Selector* expression based on the *muws-p2-xs:PreviousState*  
755 element can be used.

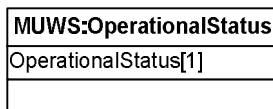
## 756 3.3 Operational Status

757 The manageability capability URI for this capability is  
758 <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/OperationalStatus>

### 759 3.3.1 Definition

760 The operational status capability defines a simple representation of the availability of a resource.  
761 This is expressed in terms defined by MUWS. These terms are independent of any specific state  
762 model, as defined by domain experts. An operational status property reflects whether the  
763 resource is available, unavailable, or degraded. Operational status does not conform to a specific  
764 state model. Rather, each value may correspond to more than one state in the operational state  
765 model, and conversely more than one operational status value may correspond to a single state  
766 in the operational state model. The manageable resource provides the appropriate mapping from  
767 state to status and sets the *OperationalStatus* property accordingly.

768 Figure 3 shows the UML representation of the *Operational Status* capability.



769

770

Figure 3: Operational Status

### 771 3.3.2 Properties

772 The operational status properties and elements are specified as follows:

```
773 <muws-p2-xs:OperationalStatus>
774 (Available|PartiallyAvailable|Unavailable|Unknown)
775 </muws-p2-xs:OperationalStatus>
```

776 The following fragment provides an example from a resource properties instance document  
777 containing this property:

```
778 <muws-p2-xs:OperationalStatus>Available</muws-p2-xs:OperationalStatus>
```

779 The *muws-p2-xs:OperationalStatus* property is of type *muws-p2-xs:OperationalStatusType*. The  
780 type is a restriction of *xs:string* and provides a simple indication of the availability of the resource,  
781 independent of the potentially complex operational state model. This property has a cardinality of  
782 1. The valid values are:

- 783 • *Available*: This value indicates that a manageable resource is operating normally within  
784 any configured operating parameters, and is able to perform all functional tasks.
- 785 • *PartiallyAvailable*: This value indicates that a manageable resource is operating, but  
786 outside of configured operating parameters. A manageable resource reporting this  
787 operational status is able to perform some, but not all, functional tasks. A manageable  
788 resource may, for example, be in the process of starting or a resource may be lacking  
789 some resource it needs to perform.
- 790 • *Unavailable*: This value indicates that a manageable resource is not operating, and is not  
791 able to perform any functional tasks. A manageable resource may have been stopped,  
792 or may have failed.
- 793 • *Unknown*: This value indicates that a manageable resource is unable to report status at  
794 this time.

795 Metadata for *OperationalStatus*:  
796 It is *Mutable*  
797 It is not *Modifiable*  
798 It has the following *Capability* metadata item:

```
799 <muws-p2-xs:Capability>  
800   http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/OperationalStatus  
801 </muws-p2-xs:Capability>
```

### 802 3.3.3 Events

803 The *muws-events:OperationalStatusCapability* topic defined below is used for events related to  
804 the *Operational Status* capability.

```
805 <wstop:Topic name="OperationalStatusCapability"  
806   messageTypes="muws-p1-xs:ManagementEvent" >  
807 </wstop:Topic>
```

808 No specific event is defined, since the notification on property value change provided by WS-  
809 ResourceProperties is sufficient, when applied to the *muws-p2-xs:OperationalStatus* property.

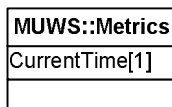
### 810 3.4 Metrics

811 The manageability capability URI for this capability is  
812 <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Metrics>

#### 813 3.4.1 Definition

814 A metric is a specific type of property. A metric represents a collected value during a collection  
815 period. A common characteristic of metrics is that they change over time. This section defines  
816 how to represent metrics and the metadata necessary to correctly process and interpret a metric  
817 value.

818 Figure 4 presents the *Metrics* capability.



819  
820

Figure 4: MUWS metrics

821 As a simple example, to clarify what a metric is, consider a toll bridge with two properties, the  
822 length of the bridge and the number of cars that have passed over the bridge. The length of the  
823 bridge, while numeric is not a metric. Length represents a current configuration of the bridge. One  
824 can not reset the length of the bridge. By contrast, the number of cars that have passed over the  
825 bridge is a metric. It requires collecting, counting, or measuring the number of cars. Typically, a  
826 count occurs for some interval, or duration of time, such as the last hour, the last day, or, since  
827 the bridge was constructed. One might reset the number of cars, for example, at the start of a  
828 new interval.

#### 829 3.4.2 Information Markup Declarations

830 The following schema fragment declares the (reusable) data type used to expose the metrics of a  
831 resource. All attributes defined in the *muws-p2-xs:MetricAttributes* attribute group are OPTIONAL.

```
832 <xs:attributeGroup name="MetricAttributes" >  
833   <xs:attribute name="ResetAt" type="xs:dateTime" />  
834   <xs:attribute name="LastUpdated" type="xs:dateTime" />  
835   <xs:attribute name="Duration" type="xs:duration" />
```

836 </xs:attributeGroup>

837 **(MetricAttributes)** attribute group MUST be included in every metric type or metric type property  
838 element declaration.

839 **(MetricAttributes)/ResetAt** indicates the time when a metric value was reset. See the definition  
840 of *muws-p2-xs:TimeScope* for information on when to provide this attribute. If the attribute value  
841 does not include a time zone indication, or Z for UTC, then the value MUST be interpreted as  
842 UTC.

843 **(MetricAttributes)/LastUpdated** indicates the last update time of a metric value. If the value  
844 does not include a time zone indication, or Z for UTC,, then the value MUST be interpreted as  
845 UTC.

846 **(MetricAttributes)/Duration** indicates the time over which a metric value was collected, counted,  
847 or measured previous to the *LastUpdated* time. The *Duration* attribute MUST be included for a  
848 metric having a *TimeScope* of *Interval* and MUST NOT be included for a metric having a  
849 *TimeScope* of *PointInTime* and *SinceReset*. For these cases, an implementer should make use of  
850 *ResetTime* and *CurrentTime* to calculate the duration for the collection of a metric value.

851 The following metric type definition is an example of how a metric attribute is incorporated into a  
852 metric type. All metric types MUST incorporate the *muws-p2-xs:MetricAttributes* attribute group.

```
853 <xs:complexType name="MyExampleIntegerMetricType">  
854   <xs:simpleContent>  
855     <xs:extension base="xs:integer">  
856       <xs:attributeGroup ref="muws-p2-xs:MetricAttributes"/>  
857       <xs:anyAttribute namespace="##other" processContents="lax"/>  
858     </xs:extension>  
859   </xs:simpleContent>  
860 </xs:complexType>
```

861 The following fragment shows an example instance of the above metric type.

```
862 <MyIntegerMetric  
863   LastUpdated="2004-03-11T11:30:56Z"  
864   Duration="PT1H">  
865   12345  
866 </MyIntegerMetric>
```

### 867 3.4.3 Metadata

868 The following metadata is applicable to any property that is a metric:

869 It is *Mutable*

870 It is not *Modifiable*

871 It has the following *Capability* metadata item:

```
872 <muws-p2-xs:Capability>  
873   http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Metrics  
874 </muws-p2-xs:Capability>
```

875 The following additional metadata items are defined for a property that is a metric:

```
876 <muws-p2-xs:ChangeType>(Counter|Gauge|Unknown)</muws-p2-xs:ChangeType>
```

877 **muws-p2-xs:ChangeType** is an enumeration indicating how a change to an associated metric  
878 value should be interpreted by a consumer. A property representing a metric MUST include a  
879 single instance of *ChangeType* in its metadata description. Each *ChangeType* value is interpreted  
880 as follows:

- 881 • *Counter* - the value of the metric is a monotonically increasing integer. Such a metric  
882 value increases by increments of "1" over successive counts, collections, or  
883 measurements.

- 884 • *Gauge* – changes of the value of the metric are not constrained in the way changes to
- 885 *Counter* metrics are constrained.
- 886 • *Unknown* - the change behavior for the value of the metric is not known or cannot be
- 887 described.

888

```
889 <muws-p2-xs:TimeScope>
890   ( Interval | PointInTime | SinceReset )
891 </muws-p2-xs:TimeScope>
```

892 **muws-p2-xs:TimeScope** is an enumeration for indicating if there is some interval, over which the  
 893 data is collected, counted, or measured. A property that is a metric MUST include a single  
 894 instance of *TimeScope* in its metadata description. Each *TimeScope* value is interpreted as  
 895 follows:

- 896 • *Interval* - the value of a metric is collected over some time interval. In this case a *Duration*
- 897 attribute MUST be reported with a metric property. The value of a *Duration* attribute is the
- 898 elapsed time, from the beginning of an interval, to the end of an interval. A *Duration*
- 899 usually remains the same for every reading of a metric. The *ResetAt* attribute MAY also
- 900 be reported with such a metric property.
- 901 • *PointInTime* - the value of a metric is counted, collected, or measured at a single instant
- 902 in time. In this case a *Duration* attribute MUST NOT be reported with a metric property.
- 903 A metric defined with a *TimeScope* of *PointInTime* does not support a reset capability
- 904 and MUST NOT include a *ResetAt* attribute.
- 905 • *SinceReset* - the value of the metric is collected since the last reset of a resource, or
- 906 since the manageable resource started collecting data for a metric. . In this case a
- 907 *Duration* attribute MUST NOT be reported with a metric property, and a *ResetAt* attribute
- 908 MUST be reported.

909

```
910 <muws-p2-xs:gatheringTime>
911   ( OnChange | Periodic | OnDemand | Unknown )
912 </muws-p2-xs:gatheringTime>
```

913 **muws-p2-xs:GatheringTime** is an enumeration indicating under which circumstance the value of  
 914 a metric is updated. A property that is a metric MUST include a single instance of *muws-p2-*  
 915 *xs:GatheringTime* in its metadata description. Each *muws-p2-xs:GatheringTime* value is  
 916 interpreted as follows:

- 917 • *OnChange* - the value of a metric is updated whenever a change occurs to the quantity
- 918 measured.
- 919 • *Periodic* - the value of a metric is updated on a regularly scheduled basis.
- 920 • *OnDemand* - the value of a metric is updated when processing a request for the metric
- 921 value.
- 922 • *Unknown* - it is unknown when the value of a metric is updated.

923

```
924 <muws-p2-xs:CalculationInterval>xs:duration</muws-p2-
925 xs:CalculationInterval>
```

926 **muws-xs-p2:CalculationInterval** represents the interval at which a value of a metric is gathered  
 927 or calculated by a resource. The value of a metric is not updated during a calculation interval.  
 928 Unlike *Duration*, which can change every time the metric is updated, the value of  
 929 *CalculationInterval* is expected to change rarely. This is because *CalculationInterval* is used only  
 930 for a value of a metric that is updated at regular intervals.

931

```
932 <muws-p2-xs:MetricGroup>xs:anyURI</muws-p2-xs:MetricGroup>
```



933 **muws-p2-xs:MetricGroup** indicates that a metric property is a member of a group of metrics. A  
934 metric property MAY be a member of zero or more metric groups. A metric group is identified by a  
935 URI. Each metric property included in a metric group MUST have a *muws-p2-xs:MetricGroup*  
936 element containing an identical URI. A metric property MAY include zero or more *muws-p2-*  
937 *xs:MetricGroup* elements in its metadata description. Each *muws-p2-xs:MetricGroup* element  
938 represents a membership of the metric property in a metric group.

### 939 3.4.4 Properties

940 The following fragment provides the specification of a resource metrics property:

```
941 <muws-p2-xs:CurrentTime>xs:dateTime</muws-p2-xs:CurrentTime>
```

942 **muws-p2-xs:CurrentTime** contains the current time, as known to a resource, when a property  
943 was retrieved from a manageable resource. This property is useful to a manageability consumer,  
944 in the absence of a time synchronization mechanism, when analyzing the time values received  
945 from a manageability endpoint. *muws-p2-xs:CurrentTime* is a read-only mandatory property with  
946 a resource cardinality of 1.

947 The Metrics capability requires the *muws-p2-xs:CurrentTime* property to be present in a resource  
948 property. The *muws-p2-xs:CurrentTime* property provides a reference point for time-based  
949 attributes, as defined by metric data types. Note that *muws-p2-xs:CurrentTime* is not a metric.  
950 Rather, it is a property of type *xs:dateTime* defined as part of the "Metrics" capability,  
951 consequently, any reset operations has no effect on *muws-p2-xs:CurrentTime*.

### 952 3.4.5 Events

953 The *muws-events:MetricsCapability* topic defined below is used for events related to the *Metrics*  
954 capability.

```
955 <wstop:Topic name="MetricsCapability"  
956           messageTypes="muws-pl-xs:ManagementEvent ">  
957 </wstop:Topic>
```

958 *WS-ResourceProperties* specifies the ability to define optional topics for a resource property that  
959 can emit notifications when a value changes. These topics allow a consumer to request  
960 notifications on an update of a metric property.

## 961 3.5 Configuration

962 The manageability capability URI for this capability is  
963 <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Configuration>

### 964 3.5.1 Definition

965 A configuration property is any resource property exposing a value that, when changed, changes  
966 some operational behavior of the resource.

967 The value of a configuration property may be changed directly by a set operation, or, may be  
968 changed as a side effect of some other operation.

### 969 3.5.2 Properties

970 MUWS does not define any required property for the *Configuration* capability. Domain experts  
971 can define configuration properties which are then marked as associated with the configuration  
972 capability. The metadata for a configuration property MUST be:

973 It is *Mutable*

974 It is *Modifiable* only if the *WS-ResourceProperties SetResourceProperty* operation can be used to  
975 change the value of the property. It is not *Modifiable* if the property is changed only as a side

976 effect.  
977 It has the following *Capability* metadata item:

```
978 <muws-p2-xs:Capability>  
979     http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Configuration  
980 </muws-p2-xs:Capability>
```

### 981 **3.5.3 Operations**

982 WS- ResourceProperties *SetResourceProperty* operation MAY be used to change a configuration  
983 value.

### 984 **3.5.4 Events**

985 The *muws-events:ConfigurationCapability* topic defined below is used for events related to the  
986 *Configuration* capability.

```
987 <wstop:Topic name="ConfigurationCapability"  
988     messageTypes="muws-pl-xs:ManagementEvent ">  
989 </wstop:Topic>
```

---

## 4 Capabilities applicable to management in general

990

991

992 Section 3, "Capabilities applicable to manageable resources", when merged with the capabilities  
993 defined in [MUWS Part 1], provide the list of manageability capabilities defined by MUWS. This  
994 section provides management-related capabilities that are different from manageability  
995 capabilities.

996 A *manageability capability* is offered by a manageability representation and a manageability  
997 capability applies to a resource as represented by a manageability representation. In contrast, a  
998 *management-related capability* can be offered by any endpoint of a Web service, not just a  
999 manageability endpoint.

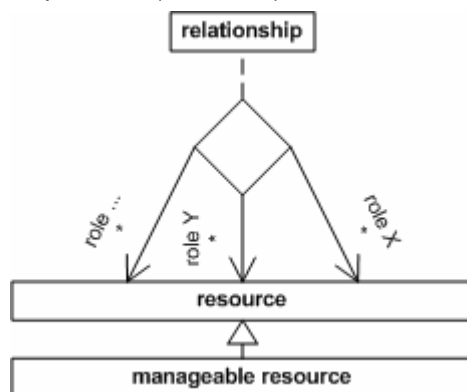
1000 The function of a management-related capability is related to the management of a resource, but  
1001 it is not necessarily offered directly by a manageability endpoint of a resource. For example, the  
1002 capability to help a manageability consumer discover a new manageable resource can be  
1003 provided by a registry instead of by a management representation of the resource. As another  
1004 example, a manageable resource may provide information about relationships in which it  
1005 participates. The information about a relationship may also provide valid information for another  
1006 entity or resource that is not manageable, like a registry, maintaining and providing relationship  
1007 information about a resource without the resource providing the relationship information directly.

### 4.1 Relationships

1009 The manageability capability URI for this capability is  
1010 <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Relationships>

#### 4.1.1 Definition

1012 A relationship is an N-ary association between resources. A relationship may have properties and  
1013 other characteristics. One of these properties is a type that conveys the semantic of the  
1014 relationship. The resources involved in the relationship are called participants. Each participant  
1015 has a role in the relationship. The participants may or may not be manageable resources in the  
1016 MUWS sense. The notion of "direction" of a relationship is a semantic interpretation based on role  
1017 definitions. There could be many instances of relationships between many instances of  
1018 resources. The arrows in Figure 5 depict navigability, which means that by following the arrow  
1019 one could resolve what the end points to (reference).

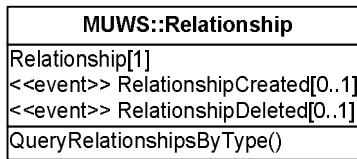


1020

1021

Figure 5: Relationship conceptual model

1022 Note that this capability is not limited to manageable resources and can be exposed by any  
 1023 resource that wants to expose relationships that it knows about.  
 1024 Figure 6 is a UML representation of the relationship capability.



1025  
 1026

Figure 6: Relationship capability

1027 A relationships may become stale. The information about a relationship should be validated,  
 1028 either manually or automatically, before it can be relied upon. Exposing the information about a  
 1029 relationship should be considered a potential security risk if a participating resource should not be  
 1030 visible for security reasons.

## 1031 4.1.2 Information Markup Declarations

### 1032 4.1.2.1 Representation of Categories of Relationships

1033 A relationship may be categorized as a certain type of relationship. A relationship type defines the  
 1034 semantics of the relationship. One relationship type may be a specialization or generalization of  
 1035 another type..This defines a taxonomy of relationship categories. MUWS defines a way to  
 1036 represent a type and its taxonomy lineage, but the actual definition of a relationship type is  
 1037 specific to a resource management model. Therefore, no relationship type is defined by MUWS.  
 1038 In other words, MUWS specifies only the mechanism to convey a relationship type, or category,  
 1039 in XML as follows.

1040 *RelationshipTypeType* type is declared as follows

```
1041 <xs:complexType name="RelationshipTypeType">
1042   <xs:complexContent>
1043     <xs:extension base="muws-p2-xs:CategoryType" />
1044   </xs:complexContent>
1045 </xs:complexType>
```

1046 The *RelationshipTypeType* type is used to declare an XML element containing instances of  
 1047 relationship type information.

1048 The relationship type information MUST be declared as follows:

- 1049 • An XML element declaring which QName identifies the semantics of a relationship type..
- 1050 • The XML element MUST be declared with an XML Schema type that is a restriction of  
 1051 *RelationshipTypeType*.
- 1052 • The contents of the XML element MUST be either
  - 1053 • The only one XML element corresponding to the generalization of the currently  
 1054 declared relationship type
  - 1055 • The empty sequence, if the currently declared relationship type does not have a  
 1056 generalization, such as the top of a taxonomy.

1057 For example, the "USB attached" relationship type may be generalized to the "Bus connected"  
 1058 type which, in turn, may be generalized to the "Generally linked" type. An instance of the "USB  
 1059 attached" relationship type information may be represented in the following XML fragment by  
 1060 using the rules described above:

```
1061 <my:RelationshipTypeInstanceElement xsi:type="RelationshipTypeType">
1062   <usb:Attached>
1063     <bus:Connected>
1064       <generally:Linked/>
```

1065  
1066  
1067

```
<bus:Connected>  
</usb:Attached>  
</my:RelationshipTypeInstanceElement>
```

#### 1068 4.1.2.2 Representation of an Instance of a Relationship

1069 MUWS defines the following Global Element Declaration (GED) to represent an instance of a  
1070 relationship.

1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085

```
<muws-p2-xs:Relationship>  
  <muws-p2-xs:Name>xs:string</muws-p2-xs:Name> ?  
  <muws-p2-xs:Type>muws-p2-xs:RelationshipTypeType</muws-p2-xs:Type>  
  <muws-p2-xs:Participant>  
    <muws-p1-xs:ManageabilityEndpointReference/> *  
    <muws-p1-xs:ResourceId/> ?  
    <muws-p2-xs:Role>xs:anyURI</muws-p2-xs:Role>  
    {any} *  
  </muws-p2-xs:Participant>  
  <muws-p2-xs:Participant/>+  
  <muws-p2-xs:AccessEndpointReference>  
    wsa:EndpointReferenceType  
  </muws-p2-xs:AccessEndpointReference>?  
  {any} *  
</muws-p2-xs:Relationship>
```

1086 **muws-p2-xs:Relationship/muws-p2-xs:Name** is a human readable name for a relationship.  
1087 *Name* should not be used for machine reasoning about the semantics of a relationship. Type  
1088 should be used instead. This element is OPTIONAL.

1089 **muws-p2-xs:Relationship/muws-p2-xs:Type** is the relationship type this relationship belongs  
1090 to. For example, linkage, containment, or dependency. .MUWS does not define any specific  
1091 relationship type. This is left to domain-specific models. MUWS only defines a way to convey the  
1092 type as part of the representation of a relationship. In order to allow relationships to be defined as  
1093 part of a taxonomy, the mechanism used by MUWS to represent relationship types leverages the  
1094 *muws-p2-xs:CategoryType* type defined in section 2.6. This element is REQUIRED.

1095 **muws-p2-xs:Relationship/muws-p2-xs:Participant** contains information about a participant in  
1096 the relationship. There MUST be at least two participants, but there MAY be more than two  
1097 participants.

1098 **muws-p2-xs:Relationship/muws-p2-xs:Participant/muws-p1-  
1099 xs:ManageabilityEndpointReference** is a reference to a WSDM manageability endpoint. This  
1100 GED is defined in part 1. It MAY be included if a participant is a WSDM manageable resource  
1101 and the provider wishes to expose this information.. If more than one manageability endpoint is  
1102 known, then more than one instance of this element MAY be present.

1103 **muws-p2-xs:Relationship/muws-p2-xs:Participant/muws-p1-xs:ResourceID** is a WSDM  
1104 manageable resource identifier which MAY be reported by the provider of relationship  
1105 information. This GED is defined in part 1. This information may be used to locate manageability  
1106 endpoints for a participant, or may be used for other purposes. For example, a resource identifier  
1107 SHOULD be used to express that the provider of relationship information is also a participant in a  
1108 relationship by returning its own resource identifier as one of the participants. Obviously, in order  
1109 for this assertion to work, the provider of relationship information must be a WSDM manageable  
1110 resource.

1111 **muws-p2-xs:Relationship/muws-p2-xs:Participant/muws-p2-xs:Role** is a URI which identifies  
1112 the role a participant plays in a relationship. A participant role MUST be unique within a given  
1113 instance of the relationship. The set of valid roles is defined by a relationship type. This attribute  
1114 is REQUIRED.

1115 **muws-p2-xs:Relationship/muws-p2-xs:Participant/{any}\*** is an XML extensibility content  
1116 which MAY contain elements that further or otherwise describe a participant. For example, when

1117 a participant is an endpoint of a Web service, an *EndpointReference* element as defined by  
 1118 MOWS MAY be included in the extensibility content to reference a functional or operational  
 1119 endpoint of a Web service that participates in a relationship.

1120 **muws-p2-xs:Relationship/muws-p2-xs:AccessEndpoint** is a reference to a Web service  
 1121 endpoint which provides access to this relationship (if available). The endpoint MUST implement  
 1122 the relationship access capability (see section 4.2).

1123 The following is an example of a relationship information instance. The relationship is a WSDM  
 1124 manageable network host myhost.myorg.org containing an attached SCSI disk. The SCSI disk is  
 1125 not manageable by itself, but is exposed as a functional or operational endpoint of a Web service  
 1126 (e.g. to read/write from the disk). The “containment” relationship is represented by the following  
 1127 XML instance fragment:

```

1128 <muws-p2-xs:Relationship>
1129   <muws-p2-xs:Name>SCSI disk attached to the host computer</muws-p2-
1130   xs:Name>
1131   <muws-p2-xs:Type>
1132     <scsi:Attached>
1133       <bus:Connected>
1134         <generally:Linked/>
1135       </bus:Connected>
1136     </scsi:Attached>
1137   </muws-p2-xs:Type>
1138   <muws-p2-xs:Participant>
1139     <muws-p1-xs:ManageabilityEndpointReference>
1140       ...EPR1...
1141     </muws-p1-xs:ManageabilityEndpointReference>
1142     <muws-p1-xs:ResourceID>urn:uuid:123</muws-p1-xs:ResourceID>
1143     <muws-p2-xs:Role>urn:role:bus:host</muws-p2-xs:Role>
1144     <netop-xs:HostName>myhost.myorg.org</netop-xs:NostName>
1145   </muws-p2-xs:Participant>
1146   <muws-p2-xs:Participant>
1147     <muws-p2-xs:Role>urn:role:bus:device</muws-p2-xs:Role>
1148     <scsi-xs:Port>2</scsi-xs:Port>
1149     <scsi-xs:CH>0</scsi-xs:CH>
1150     <scsi-xs:BusID>5</scsi-xs:BusID>
1151     <scsi-xs:LUN>0</scsi-xs:LUN>
1152     <mows-xs:EndpointReference>
1153       ...EPR2...
1154     </mows-xs:EndpointReference>
1155   </muws-p2-xs:Participant>
1156 </muws-p2-xs:Relationship>
  
```

### 1157 4.1.3 Properties

1158 The Relationship capability defines the following property:

```

1159 <muws-p2-xs:Relationship/> *
  
```

1160 **muws-p2-xs:Relationship** is a representation of a relationship of which the provider of this  
 1161 capability is aware. See section 4.1.2 for the definition of the Relationship element. The provider  
 1162 of this capability is not necessarily a participant in any relationship represented by this property.

1163 It is not recommended to request all values of the Relationship property with either *wsrf-*  
 1164 *rp:GetResourceProperty* or *wsrf-rp:GetMultipleResourceProperties* operations as there may be  
 1165 too many relationships. The use of the *wsrf-rp:QueryResourceProperties* operation is  
 1166 RECOMMENDED when retrieving the Relationships property. A provider of this manageability  
 1167 capability SHOULD, in general, support the *wsrf-rp:QueryResourceProperties* operation.  
 1168 However, if the provider of this capability knows of just a few relationships, it MAY choose not to  
 1169 support *wsrf-rp:QueryResourceProperties* operation.

1170 For example, the following request may be sent to retrieve all “Bus connected” relationships  
1171 which point to devices exposed as Web services.

```
1172 <soap:Envelope ...>  
1173   <soap:Header>  
1174     ...  
1175   </soap:Header>  
1176   <soap:Body>  
1177     <wsrf-rp:QueryResourceProperties>  
1178       <wsrf-rp:QueryExpression  
1179         Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116" >  
1180 boolean(/*/muws-p2-xs:Relationship/muws-p2-xs:Type/*bus:Connected and  
1181 /*/muws-p2-xs:Relationship/muws-p2-  
1182 xs:Participant[Role="urn:role:bus:device"]/mows-xs:EndpointReference)  
1183       </wsrf-rp:QueryExpression>  
1184     </wsrf-rp:QueryResourceProperties>  
1185   </soap:Body>  
1186 </soap:Envelope>
```

## 1187 4.1.4 Operations

1188 This capability defines the following message exchanges.

### 1189 4.1.4.1 QueryRelationshipsByType

1190 This operation is OPTIONAL. It is a shortcut to query relationships of the same type. The request  
1191 to perform this operation has a payload as follows:

```
1192 <muws-p2-xs:QueryRelationshipsByType>  
1193   <muws-p2-xs:RequestedType>xs:QName/muws-p2-xs:RequestedType > +  
1194 </muws-p2-xs:QueryRelationshipsByType>
```

1195 **muws-p2-xs:QueryRelationshipsByType** is a Global Element Declaration (GED) which  
1196 identifies the operation requested.

1197 **muws-p2-xs:QueryRelationshipsByType/muws-p2-xs:RequestedType** is a QName which  
1198 identifies the requested type(s) of relationship(s). When processing this request, the  
1199 manageability endpoint MUST return any available instance relationship that is of the requested  
1200 type or of any type that is a specialization of the requested type. There can be more than one  
1201 requested type, in which case any relationship instance corresponding to any requested type  
1202 MUST be returned.

1203 The response to the above request is either a fault (any fault) or the following message:

```
1204 <muws-p2-xs:QueryRelationshipsByTypeResponse>  
1205   <muws-p2-xs:Relationship/> *  
1206 </muws-p2-xs:QueryRelationshipsByTypeResponse>
```

1207 **muws-p2-xs:QueryRelationshipsByTypeResponse** is a GED which identifies a response to  
1208 the requested operation.

1209 **muws-p2-xs:QueryRelationshipByTypeResponse/muws-p2-xs:Relationship** is a relationship  
1210 representation matching a requested type. There is one such element for each relationship  
1211 instance corresponding to at least one requested type.

1212 This operation has the following *Capability* metadata item:

```
1213 <muws-p2-xs:Capability>  
1214   http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Relationships  
1215 </muws-p2-xs:Capability>
```



## 1216 4.1.5 Events

1217 To support notifications on a change in a relationship, the following notification topics are defined  
1218 in the manageable relationships capability:

```
1219 <wstop:Topic name="RelationshipCreated" messageTypes="muws-p2-  
1220 xs:RelationshipCreatedNotification" >  
1221   <wstop:MessagePattern  
1222     Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">//muws-p1-  
1223 xs:ManagementEvent[count(muws-p2-xs:RelationshipCreatedNotification)=1]  
1224   </wstop:MessagePattern>  
1225 </wstop:Topic>  
1226 <wstop:Topic name="RelationshipDeleted" messageTypes="muws-p2-  
1227 xs:RelationshipDeletedNotification" >  
1228   <wstop:MessagePattern  
1229     Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">//muws-p1-  
1230 xs:ManagementEvent[count(muws-p2-xs:RelationshipDeletedNotification)=1]  
1231   </wstop:MessagePattern>  
1232 </wstop:Topic>
```

1233 **muws-events:ManageableRelationships/muws-events:RelationshipCreated** indicates the  
1234 addition of a new relationship. It is RECOMMENDED that a consumer subscribe to this  
1235 notification with an appropriate selector against the content of notification messages in order to  
1236 reduce the volume of received messages. Each notification message contains at least the  
1237 following information:

```
1238 <RelationshipCreatedNotification>  
1239   <Relationship/>  
1240 </RelationshipCreatedNotification>
```

1241 **muws-events:ManageableRelationships/muws-events:RelationshipDeleted** indicates  
1242 removal of an existing relationship. It is RECOMMENDED that a consumer subscribe to this  
1243 notification with an appropriate selector against the content of notification messages in order to  
1244 reduce the volume of received messages. Each notification message contains at least the  
1245 following information:

```
1246 <RelationshipDeletedNotification>  
1247   <Relationship/>  
1248 </RelationshipDeletedNotification>
```

## 1249 4.2 Relationship Access Capability

1250 The manageability capability URI for this capability is  
1251 <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/RelationshipAccess>

### 1252 4.2.1 Definition

1253 Sometimes, a relationship is more than just a reflection of some physical fact. A relationship may  
1254 also have its own properties, operations, events, and lifecycle. In this case, interactions with a  
1255 relationship service could cause, as a side effect, a system or physical fact to be changed in  
1256 order to comply with the semantics of its role in the relationship. For this reason, we allow a  
1257 relationship to be exposed as an independent service. The provider of a Web service endpoint  
1258 supporting the *Relationship Access* capability also provides access to the participants in a  
1259 relationship. If this capability is supported, then an endpoint reference for a service implementing  
1260 the capability MUST contain sufficient information to allow a provider to disambiguate which  
1261 relationship is being accessed by a message exchange. An endpoint reference could be obtained  
1262 from the *muws-p2-xs:Relationship/AccessEndpointReference* in relationship element defined in  
1263 section 4.1.2.2.

1264 The endpoint in this case is a WS-Resource, not a WSDM Manageable resource. Section 4.2.2  
1265 describes relationships as WSDM Manageable resources. The relationship access endpoint  
1266 supports any exchange of messages where the exchange is specific to a particular relationship  
1267 and management model, and, where the exchange is necessary in order to provide access to the  
1268 relationship.

1269 The only other normative requirement is that if the relationship lifecycle is exposed by a provider  
1270 of this capability, then the Web service endpoint MUST implement the WS-ResourceLifetime  
1271 specification [WS-RL].

## 1272 4.2.2 Events

1273 The *muws-events:RelationshipAccessCapability* topic defined below is used for events related to  
1274 the *Relationship Access* capability.

```
1275 <wstop:Topic name="RelationshipAccessCapability"  
1276           messageTypes="muws-pl-xs:ManagementEvent">  
1277 </wstop:Topic>
```

## 1278 4.3 Relationship Resource Capability

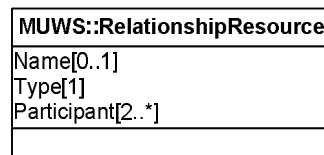
1279 The manageability capability URI for this capability is  
1280 <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/RelationshipResource>

### 1281 4.3.1 Definition

1282 A Web service endpoint, in addition to providing access to a relationship as described in section  
1283 4.2, may also represent a relationship. Representing a relationship means that an endpoint is  
1284 able to provide relationship information as described in section 4.1.2.2. In this case, a Web  
1285 service endpoint MUST be a WS-Resource, as defined by the WSRF. One such WS-Resource  
1286 provides information about one relationship instance. Representing a relationships as WS-  
1287 Resource is useful when a manageability model defines additional properties, operations or  
1288 events for a relationship.

1289 In order to represent a relationship as a WS-Resource, a set of properties is normatively required.  
1290 The rest of the representation depends upon the relationship manageability model and discretion  
1291 of the provider of a WS-Resource and relationship.

1292 Figure 7 is a UML representation of the Relationship Resource capability.



1293

1294

Figure 7: Relationship Resource capability

### 1295 4.3.2 Properties

1296 The Relationship Resource capability defines the following properties.

```
1297 <muws-p2-xs:Name>xs:string</muws-p2-xs:Name> ?
```

1298 **muws-p2-xs:Name** is an element as defined by the Relationship/Name in section 4.1.2.2. It is  
1299 OPTIONAL.

1300

```
1301 <muws-p2-xs:Type>muws-p2-xs:RelationshipTypeType</muws-p2-xs:Type>
```

1302 **muws-p2-xs:Type** is an element as defined by the Relationship/Type in section 4.1.2.2. It is  
1303 REQUIRED and can only appear once.

1304

```
1305 <muws-p2-xs:Participant>  
1306   <muws-p1-xs:ManageabilityEndpointReference/> *  
1307   <muws-p1-xs:ResourceId/> ?  
1308   <muws-p2-xs:Role>xs:anyURI</muws-p2-xs:Role>  
1309   {any} *  
1310 </muws-p2-xs:Participant>
```

1311 **muws-p2-xs:Participant** is an element as defined by the Relationship/Participant in section  
1312 4.1.2.2. This element MUST appear at least twice, and exactly once per participant in the  
1313 relationship.

### 1314 4.3.3 Events

1315 The *muws-events:RelationshipResourceCapability* topic defined below is used for events related  
1316 to the *Relationship Resource* capability.

```
1317 <wstop:Topic name="RelationshipResourceCapability"  
1318             messageTypes="muws-p1-xs:ManagementEvent ">  
1319 </wstop:Topic>
```

## 1320 4.4 Advertisement

1321 The manageability capability URI for the Advertisement capability is  
1322 <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Advertisement>

### 1323 4.4.1 Definition

1324 The *Advertisement* capability is exposed by a Web service that is able to provide a notification on  
1325 the creation or the destruction of a manageable resource. Since a consumer cannot register for a  
1326 notification on a resource before the resource is created, a creation event is reported for some  
1327 other resource by the implementer of a “lifetime notification” capability. .

1328 Note that this capability may be implemented by a manageable resource or by some other  
1329 service (see section 4 on the distinction between “manageability capability” and “management-  
1330 related capability”). A service might offer a capability to notify on the creation or the destruction of  
1331 a resource even though the service itself is not manageable. For example, if a system includes a  
1332 registry, to which a resource is added as soon as it is created, and from which it is removed when  
1333 it is destroyed, then this registry could expose the *Advertisement* capability and use it to share  
1334 information about resource creation and destruction events with manageability consumers.  
1335 Likewise, a resource factory might emit creation events for a resource it creates, yet the factory  
1336 itself might not be manageable. Another example is a container, a J2EE server or a business  
1337 process execution engine for example, that can send a notification when a contained resource is  
1338 created.

1339 This capability defines four topics used for notification but does not define any property or  
1340 operation.

1341 In addition to advertisement by sending notifications, as defined in this capability, another  
1342 approach for advertisement is to register a manageable resource in a registry. A resource  
1343 advertised in this way can be discovered using the mechanisms introduced in section 5.2.

1344 Figure 8 is a UML representation of the *Advertisement* capability.

MUWS::Advertisement
<<event>> ManageabilityEndpointCreation[0..1]
<<event>> ManageableResourceCreation[0..1]
<<event>> ManageabilityEndpointDestruction[0..1]
<<event>> ManageableResourceDestruction[0..1]

1345

1346

Figure 8: Advertisement capability

## 1347 4.4.2 Events

1348 The Advertisement capability defines four notification topics:

```

1349 <wstop:Topic name="ManageabilityEndpointCreation" messageTypes="muws-p2-
1350 xs:CreationNotification">
1351   <wstop:MessagePattern
1352     Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">//muws-p1-
1353 xs:ManagementEvent[count(muws-p2-xs:CreationNotification)=1]
1354   </wstop:MessagePattern>
1355   <wstop:Topic name="ManageableResourceCreation" messageTypes="muws-p2-
1356 xs:CreationNotification">
1357     <wstop:MessagePattern
1358       Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">//muws-p1-
1359 xs:ManagementEvent[count(muws-p2-xs:CreationNotification)=1]
1360     </wstop:MessagePattern>
1361   </wstop:Topic>
1362 </wstop:Topic>
1363 <wstop:Topic name="ManageabilityEndpointDestruction" messageTypes="muws-
1364 p2-xs:DestructionNotification">
1365   <wstop:MessagePattern
1366     Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">//muws-p1-
1367 xs:ManagementEvent[count(muws-p2-xs:DestructionNotification)=1]
1368   </wstop:MessagePattern>
1369   <wstop:Topic name="ManageableResourceDestruction" messageTypes="muws-p2-
1370 xs:DestructionNotification"/>
1371   <wstop:MessagePattern
1372     Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">//muws-p1-
1373 xs:ManagementEvent[count(muws-p2-xs:DestructionNotification)=1]
1374   </wstop:MessagePattern>
1375   </wstop:Topic>
1376 </wstop:Topic>

```

1377 The **"muws-events:ManageabilityEndpointCreation"** topic corresponds to notification on the  
1378 creation of a new manageability endpoint for a new or existing resource. A manageability  
1379 endpoint may be created in conjunction with, or independent of, the creation of the manageable  
1380 resource. A new manageability endpoint could be the first one for a resource or be an addition to  
1381 others. An associated *muws-p2-xs:CreationNotification* message contains the EPR of a newly  
1382 created manageability endpoint.

1383 The **"muws-events:ManageableResourceCreation"** topic is a specialization of the  
1384 "Manageability EndpointCreation" topic. This topic corresponds to the case where a resource  
1385 itself is newly created. Note that if a resource is created that is not manageable (i.e. which does  
1386 not have a manageability endpoint) no notification on this topic will be sent. If a resource and a  
1387 manageability endpoint for the resource are created then a notification will be sent to a subscriber  
1388 on this topic.

1389 The **"muws-events:ManageabilityEndpointDestruction"** topic corresponds to notification on  
1390 the destruction of a manageability endpoint. It does not imply that the associated resource was  
1391 destroyed. An associated *muws-p2-xs:DestructionNotification* message contains the *muws-p2-*  
1392 *xs:ResourceId* that a newly destroyed manageability endpoint provided for the resource before its  
1393 destruction.

1394 The “**muws-events:ManageableResourceDestruction**” topic is a specialization of the  
1395 “ManageabilityEndpointDestruction” topic. This topic corresponds to the case where a resource  
1396 itself is destroyed at the same time as the manageability endpoint. Note that if a resource is  
1397 destroyed that is not manageable (i.e. which does not have a manageability endpoint) no  
1398 notification on this topic will be sent. An associated *muws-p2-xs:DestructionNotification* message  
1399 contains the *muws-p2-xs:ResourceId* that a newly destroyed manageability endpoint provided for  
1400 the resource before its destruction.

1401 The content element for these topics are described as follows:

```
1402 <muws-p2-xs:CreationNotification">  
1403   <muws-p1-xs:ManageabilityEndpointReference"/> *  
1404 </muws-p2-xs:CreationNotification">
```

1405 **muws-p2-xs:CreationNotification/muws-p1-xs:ManageabilityEndpointReference** is a  
1406 reference to the manageability endpoint of a newly created resource. There can be more than  
1407 one such reference if there is more than one known manageability endpoint.

1408

```
1409 <muws-p2-xs:DestructionNotification">  
1410   <muws-p1-xs:ResourceId"/> ?  
1411 </muws-p2-xs:DestructionNotification">
```

1412 **muws-p2-xs:DestructionNotification/muws-p1-xs:ResourceId** is the *ResourceId* of a newly  
1413 destroyed resource.

1414

---

## 1415 5 Discovery

1416 Many forms of discovery are supported by Web services. This specification does not prescribe a  
1417 normative method for discovering manageability services. It is expected that discovery methods  
1418 commonly used for Web services will be used as discovery methods for manageability services.  
1419 The goal of discovery is to obtain the EPR of a manageability endpoint. The Advertisement  
1420 capability (section 4.4), when supported, provides one way to facilitate discovery via events. This  
1421 section also describes two other ways to discover manageable resources. These are just some of  
1422 the discovery methods that can be used.

1423 The only normative requirement relative to discovering manageability services is that a  
1424 manageability service **MUST** provide the Identity capability as defined by MUWS. As a result of  
1425 this requirement, a consumer can inspect the WSDL description for a Web service or attempt to  
1426 use the Identity capability of a Web service to determine if a discovered service acts as a  
1427 manageability service. If a discovered service provides at least the Identity capability as defined  
1428 by MUWS, then it is a manageability service.

### 1429 5.1 Discovery using Relationships

1430 There are at least two scenarios in which a relationship can be used to discover a manageable  
1431 resource.

1432 The first scenario is when a manageable resource points to some other manageable resource  
1433 through a relationship. A manageable resource that supports the Relationship capability enables  
1434 discovery of an EPR for some other resource that participates in a relationship with the  
1435 manageable resource. This is done by using the "Relationship" property defined in section 4.1.3  
1436 or invoking the operations defined in section 4.1.4. Any EPRs contained in such a response  
1437 message may be used by the manageability consumer to disambiguate a manageable resource  
1438 in an exchange of messages with a manageability endpoint.

1439 The second scenario is when a consumer has access to a WS-Resource representing a  
1440 relationship and the relationship has a manageable resource as a member. A consumer can then  
1441 use the properties of the Relationship Resource capability to retrieve any EPRs of a manageable  
1442 resource participating in the relationship.

### 1443 5.2 Discovery using Registries

1444 In addition to emitting a notification on the creation and the destruction of a resource as defined  
1445 by the Advertisement capability in section 4.4, a resource can be advertised to a registry by  
1446 invoking an insertion interface of the registry. A consumer can then discover a manageable  
1447 resource by invoking a query interface of the registry.

1448 The WSRF WS-Service Group specification [WS-SG] defines a type of registry, along with the  
1449 message exchanges used to interact with a registry of this type. It is **RECOMMENDED** that a  
1450 registry used to discover a manageable resource conforms to the WS-Service Group specification  
1451 and that the registry conform to the following additional constraints:

1452 The service group **SHOULD** include as properties the following two elements:

```
1453 <wssg:MembershipContentRule  
1454   MemberInterface="muws-pl-xs:Identity"  
1455   ContentElements="muws-pl-xs:ResourceId">  
1456 <wssg:MembershipContentRule  
1457   MemberInterface="muws-pl-xs:ManageabilityCharacteristics"  
1458   ContentElements="muws-pl-xs:ManageabilityCapability">
```

1459 The service group **MAY** also have any other "MembershipContentRule", including a rule with an  
1460 empty value for both MemberInterface and ContentElements. In effect, this lifts any constraint on

1461 a member of the service group. The two membership content rules defined above are useful even  
1462 in a service group with no effective constraint because they allow querying the service group on  
1463 the “ResourceId” and “ManageabilityCapability” properties.

1464 When adding a manageability endpoint for a resource to the membership of a service group using  
1465 the “Add” operation, the requestor SHOULD include the *muws-p1-xs:ResourceId* element of a  
1466 manageable resource in a *wssg:Add/wssg:Content* element of a request, even if the service  
1467 group supports additional membership content rules that would have permitted registration of a  
1468 manageability endpoint in the service group without providing this content element. Similarly, if  
1469 the manageable resource supports the Manageability Characteristics capability, then the  
1470 consumer SHOULD include all the *muws-p1-xs:ManageabilityCapability* elements of a  
1471 manageable resource in a *wssg:Add/wssg:Content* element of a request, even if the service  
1472 group supports additional membership content rules that would have permitted registration of the  
1473 manageability endpoint in the service group without providing this content element.

1474 Like any manageability endpoint, a manageability endpoint listed in a resource registry MUST  
1475 implement the Identity capability defined in [MUWS Part 1]. In addition, in order to facilitate  
1476 discovery, the manageability endpoint SHOULD implement the Manageability Characteristics  
1477 capability as defined in [MUWS Part 1].

1478



1479

## 6 References

1480

### 6.1 Normative

1481

**[MUWS Part 1]** William Vambenepe, *Web Services Distributed Management: Management using Web Services (MUWS 1.0) Part 1*, OASIS Committee Draft, December 2004, <http://docs.oasis-open.org/wsdm/2004/12/cd-wsdm-muws-part1-1.0.pdf>

1485

1486

**[XML1.0 3<sup>rd</sup> Edition]**

1487

Tim Bray, et al., *Extensible Markup Language (XML) 1.0 (Third Edition)*, W3C Recommendation, February 2004, <http://www.w3.org/TR/REC-xml>

1488

1489

1490

**[XML Schema Part 1]**

1491

Henry S. Thompson, et al. *XML Schema Part 1: Structures*, W3C Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-1/>

1492

1493

1494

**[XML Schema Part 2]**

1495

Paul V. Biron, et al. *XML Schema Part 2: Datatypes*, W3C Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-2/>

1496

1497

1498

**[WSDL]**

Erik Christensen, et al., *Web services Description Language (WSDL) 1.1*, W3C Note, March 2001, <http://www.w3.org/TR/wsdl>

1499

1500

1501

**[WS-Resource]**

Steve Graham, et al. *Web Service Resource 1.2 (WS-Resource)*, OASIS Working Draft, October 2004, <http://www.oasis-open.org/apps/org/workgroup/wsrp/download.php/9547/wsrp-WS-Resource-1.2-draft-01.doc>

1502

1503

1504

1505

1506

**[WS-Addressing]**

Don Box, et al., *Web services Addressing (WS-Addressing)*, W3C Member Submission, August 2004, <http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/>

1507

1508

1509

1510

**[WS-RP]**

Steve Graham, et al., *Web Services Resource Properties 1.2 (WS-ResourceProperties)*, OASIS Working Draft, June 2004, <http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-1.2-draft-04.pdf>

1511

1512

1513

1514

1515

**[XPath 1.0]**

James Clark, et al., *XML Path Language (XPath) Version 1.0*, W3C Recommendation, November 1999, <http://www.w3.org/TR/1999/REC-xpath-19991116>

1516

1517

1518

1519

**[WSN]**

Steve Graham, et al., *Web Services Base Notification 1.2 (WS-BaseNotification)*, OASIS Working Draft, June 2004, <http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf>

1520

1521

1522

1523

**[WST]**

William Vambenepe, *Web Services Topics 1.2 (WS-Topics)*, OASIS Working Draft, July 2004, <http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.pdf>

1524

1525

- 1526  
1527     **[RFC3066]**     IETF (Internet Engineering Task Force). RFC 3066: Tags for the  
1528     Identification of Languages, ed. H. Alvestrand. 2001,  
1529     <http://www.ietf.org/rfc/rfc3066.txt>  
1530  
1531     **[WS-RL]**     Latha Srinivasan, et al., Web Services Resource Lifetime 1.2 (WS-  
1532     ResourceLifetime), OASIS Working Draft, June 2004, [http://docs.oasis-](http://docs.oasis-open.org/wsr/2004/06/wsr-WS-ResourceLifetime-1.2-draft-03.pdf)  
1533     [open.org/wsr/2004/06/wsr-WS-ResourceLifetime-1.2-draft-03.pdf](http://docs.oasis-open.org/wsr/2004/06/wsr-WS-ResourceLifetime-1.2-draft-03.pdf)  
1534  
1535     **[WS-SG]**     Tom Maguire, et al., *Web Services Service Group 1.2 (WS-*  
1536     *ServiceGroup)*, OASIS Working Draft, June 2004, [http://docs.oasis-](http://docs.oasis-open.org/wsr/2004/06/wsr-WS-ServiceGroup-1.2-draft-02.pdf)  
1537     [open.org/wsr/2004/06/wsr-WS-ServiceGroup-1.2-draft-02.pdf](http://docs.oasis-open.org/wsr/2004/06/wsr-WS-ServiceGroup-1.2-draft-02.pdf)  
1538

## 1539     **6.2 Non-normative**

- 1540     **[SOAP]**     Don Box, et al., *Simple Object Access Protocol (SOAP) 1.1*, W3C Note,  
1541     May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>  
1542  
1543     **[WS-RF]**     WSRF OASIS technical committee, [http://www.oasis-](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf)  
1544     [open.org/committees/tc\\_home.php?wg\\_abbrev=wsrf](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf)  
1545  
1546     **[WSRMD]**     Steve Graham, et al., *Web Services Resource Metadata 1.0 (WS-*  
1547     *ResourceMetadataDescriptor)*, OASIS Working Draft, October 2004,  
1548     [http://www.oasis-open.org/committees/download.php/9758/wsr-WS-](http://www.oasis-open.org/committees/download.php/9758/wsr-WS-ResourceMetadataDescriptor-1.0-draft-01.PDF)  
1549     [ResourceMetadataDescriptor-1.0-draft-01.PDF](http://www.oasis-open.org/committees/download.php/9758/wsr-WS-ResourceMetadataDescriptor-1.0-draft-01.PDF)  
1550

---

## 1551 **Appendix A. Acknowledgements**

1552 The following individuals were members of the committee when the specification was approved  
1553 by the technical committee

1554 Guru Bhat, Jeff Bohren, Winston Bumpus, Nick Butler, Brian Carroll, Fred Carter, Michael  
1555 Clements, David Cox, John DeCarlo, Andreas Dharmawan, Mark Ellison, John Fuller, Paul  
1556 Lipton, Heather Kreger, Hal Lockhart, Frederico Maciel, Tom Maguire, Bryan Murray, Richard  
1557 Nikula, Mark Peel, Richard Pelavin, Homayoun Pourheidari, Warren Roberts, Karl Schopmeyer,  
1558 Igor Sedukhin, David Snelling, Thomas Studwell, William Vambenepe, Andrea Westerinen, Jim  
1559 Willits, Zhili Zhang.

---

## Appendix B. Notices

1560

1561 OASIS takes no position regarding the validity or scope of any intellectual property or other rights  
1562 that might be claimed to pertain to the implementation or use of the technology described in this  
1563 document or the extent to which any license under such rights might or might not be available;  
1564 neither does it represent that it has made any effort to identify any such rights. Information on  
1565 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS  
1566 website. Copies of claims of rights made available for publication and any assurances of licenses  
1567 to be made available, or the result of an attempt made to obtain a general license or permission  
1568 for the use of such proprietary rights by implementors or users of this specification, can be  
1569 obtained from the OASIS Executive Director.

1570 OASIS invites any interested party to bring to its attention any copyrights, patents or patent  
1571 applications, or other proprietary rights which may cover technology that may be required to  
1572 implement this specification. Please address the information to the OASIS Executive Director.

1573 Copyright © OASIS Open 2003-2004. *All Rights Reserved.*

1574 This document and translations of it may be copied and furnished to others, and derivative works  
1575 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,  
1576 published and distributed, in whole or in part, without restriction of any kind, provided that the  
1577 above copyright notice and this paragraph are included on all such copies and derivative works.  
1578 However, this document itself does not be modified in any way, such as by removing the  
1579 copyright notice or references to OASIS, except as needed for the purpose of developing OASIS  
1580 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual  
1581 Property Rights document must be followed, or as required to translate it into languages other  
1582 than English.

1583 The limited permissions granted above are perpetual and will not be revoked by OASIS or its  
1584 successors or assigns.

1585 This document and the information contained herein is provided on an "AS IS" basis and OASIS  
1586 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO  
1587 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE  
1588 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A  
1589 PARTICULAR PURPOSE.

1590

1591

1592

1593

1594

---

## Appendix C. Schemas

1595

```
1596 <?xml version="1.0" encoding="utf-8"?>
1597 <xs:schema
1598     targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-muws-
1599     part2-1.0.xsd"
1600     xmlns:muws-p2-xs="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-
1601     muws-part2-1.0.xsd"
1602     xmlns:muws-p1-xs="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-
1603     muws-part1-1.0.xsd"
1604     xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1605     xmlns:xs="http://www.w3.org/2001/XMLSchema"
1606     elementFormDefault="qualified" attributeFormDefault="unqualified">
1607
1608     <xs:import namespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-
1609     muws-part1-1.0.xsd"
1610             schemaLocation="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-
1611     wsdm-muws-part1-1.0.xsd"/>     <xs:import
1612     namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1613
1614     schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
1615     <xs:import namespace="http://www.w3.org/XML/1998/namespace"
1616             schemaLocation="http://www.w3.org/2001/xml.xsd"/>
1617
1618     <xs:complexType name="LangString">
1619         <xs:simpleContent>
1620             <xs:extension base="xs:string">
1621                 <xs:attribute ref="xml:lang" use="required"/>
1622                 <xs:anyAttribute namespace="##other"/>
1623             </xs:extension>
1624         </xs:simpleContent>
1625     </xs:complexType>
1626
1627
1628     <!-- Begin properties for the Description capability -->
1629     <xs:element name="Caption" type="muws-p2-xs:LangString"/>
1630     <xs:element name="Description" type="muws-p2-xs:LangString"/>
1631     <xs:element name="Version" type="xs:string"/>
1632     <!-- End properties for the Description capability -->
1633
1634     <xs:complexType name="DescriptionPropertiesType">
1635         <xs:sequence>
1636             <xs:element ref="muws-p2-xs:Caption"
1637                 minOccurs="0" maxOccurs="unbounded"/>
1638             <xs:element ref="muws-p2-xs:Description"
1639                 minOccurs="0" maxOccurs="unbounded"/>
1640             <xs:element ref="muws-p2-xs:Version"
1641                 minOccurs="0"/>
1642         </xs:sequence>
1643     </xs:complexType>
1644
1645     <xs:element name="DescriptionProperties"
1646             type="muws-p2-xs:DescriptionPropertiesType"/>
1647
1648     <xs:complexType name="CategoryType">
1649         <xs:sequence>
1650             <xs:any minOccurs="0"
1651                 namespace="##any" processContents="lax"/>
1652         </xs:sequence>
1653     </xs:complexType>
1654
```

```

1655 <xs:complexType name="StateType">
1656   <xs:complexContent>
1657     <xs:extension base="muws-p2-xs:CategoryType"/>
1658   </xs:complexContent>
1659 </xs:complexType>
1660
1661 <xs:element name="State" type="muws-p2-xs:StateType"/>
1662
1663 <xs:element name="EnteredState" type="muws-p2-xs:StateType"/>
1664 <xs:element name="PreviousState" type="muws-p2-xs:StateType"/>
1665
1666 <xs:complexType name="StateTransitionType">
1667   <xs:sequence>
1668     <xs:element ref="muws-p2-xs:EnteredState"/>
1669     <xs:element ref="muws-p2-xs:PreviousState"
1670       minOccurs="0"/>
1671     <xs:any minOccurs="0" maxOccurs="unbounded"
1672       namespace="##other" processContents="lax"/>
1673   </xs:sequence>
1674   <xs:attribute name="TransitionIdentifier" type="xs:anyURI"
1675     use="optional"/>
1676   <xs:attribute name="Time" type="xs:dateTime" use="required"/>
1677   <xs:anyAttribute namespace="##other"/>
1678 </xs:complexType>
1679
1680 <xs:element name="StateTransition"
1681   type="muws-p2-xs:StateTransitionType"/>
1682
1683
1684 <!-- Begin properties for the OperationalStatus capability -->
1685 <xs:element name="OperationalStatus">
1686   <xs:simpleType>
1687     <xs:restriction base="xs:string">
1688       <xs:enumeration value="Available"/>
1689       <xs:enumeration value="PartiallyAvailable"/>
1690       <xs:enumeration value="Unavailable"/>
1691       <xs:enumeration value="Unknown"/>
1692     </xs:restriction>
1693   </xs:simpleType>
1694 </xs:element>
1695 <!-- End properties for the OperationalStatus capability -->
1696
1697 <xs:complexType name="OperationalStatusPropertiesType">
1698   <xs:sequence>
1699     <xs:element ref="muws-p2-xs:OperationalStatus"/>
1700   </xs:sequence>
1701 </xs:complexType>
1702
1703 <xs:element name="OperationalStatusProperties"
1704   type="muws-p2-xs:OperationalStatusPropertiesType"/>
1705
1706 <xs:attributeGroup name="MetricAttributes">
1707   <xs:attribute name="ResetAt" type="xs:dateTime"/>
1708   <xs:attribute name="LastUpdated" type="xs:dateTime"/>
1709   <xs:attribute name="Duration" type="xs:duration"/>
1710 </xs:attributeGroup>
1711
1712 <!-- Begin properties for the Metrics capability -->
1713 <xs:element name="CurrentTime" type="xs:dateTime"/>
1714 <!-- End properties for the Metrics capability -->
1715
1716 <xs:complexType name="MetricsPropertiesType">
1717   <xs:sequence>

```

```

1718     <xs:element ref="muws-p2-xs:CurrentTime" />
1719   </xs:sequence>
1720 </xs:complexType>
1721
1722 <xs:element name="MetricsProperties"
1723           type="muws-p2-xs:MetricsPropertiesType" />
1724
1725 <xs:complexType name="RelationshipTypeType">
1726   <xs:complexContent>
1727     <xs:extension base="muws-p2-xs:CategoryType" />
1728   </xs:complexContent>
1729 </xs:complexType>
1730
1731 <xs:complexType name="RelationshipParticipantType">
1732   <xs:sequence>
1733     <xs:element ref="muws-p1-xs:ManageabilityEndpointReference"
1734               minOccurs="0" maxOccurs="unbounded" />
1735     <xs:element ref="muws-p1-xs:ResourceId"
1736               minOccurs="0" />
1737     <xs:element name="Role" type="xs:anyURI" />
1738     <xs:any minOccurs="0" maxOccurs="unbounded"
1739           namespace="##other" processContents="lax" />
1740   </xs:sequence>
1741   <xs:anyAttribute namespace="##other" />
1742 </xs:complexType>
1743
1744 <!-- Begin properties for the RelationshipResource capability -->
1745 <xs:element name="Name" type="xs:string" />
1746 <xs:element name="Type" type="muws-p2-xs:RelationshipTypeType" />
1747 <xs:element name="Participant"
1748           type="muws-p2-xs:RelationshipParticipantType" />
1749 <!-- End   properties for the RelationshipResource capability -->
1750
1751 <xs:complexType name="RelationshipType">
1752   <xs:sequence>
1753     <xs:element ref="muws-p2-xs:Name"
1754               minOccurs="0" />
1755     <xs:element ref="muws-p2-xs:Type" />
1756     <xs:element ref="muws-p2-xs:Participant"
1757               minOccurs="2" maxOccurs="unbounded" />
1758     <xs:element name="AccessEndpointReference"
1759               type="wsa:EndpointReferenceType" minOccurs="0" />
1760     <xs:any minOccurs="0" maxOccurs="unbounded"
1761           namespace="##other" processContents="lax" />
1762   </xs:sequence>
1763   <xs:anyAttribute namespace="##other" />
1764 </xs:complexType>
1765
1766 <!-- Begin properties for the Relationship capability -->
1767 <xs:element name="Relationship"
1768           type="muws-p2-xs:RelationshipType" />
1769 <!-- End   properties for the Relationship capability -->
1770
1771 <xs:complexType name="RelationshipPropertiesType">
1772   <xs:sequence>
1773     <xs:element ref="muws-p2-xs:Relationship"
1774               minOccurs="0" maxOccurs="unbounded" />
1775   </xs:sequence>
1776 </xs:complexType>
1777
1778 <xs:element name="RelationshipProperties"
1779           type="muws-p2-xs:RelationshipPropertiesType" />
1780

```



```

1781 <xs:element name="RelationshipCreatedNotification">
1782   <xs:complexType>
1783     <xs:sequence>
1784       <xs:element ref="muws-p2-xs:Relationship"/>
1785       <xs:any minOccurs="0" maxOccurs="unbounded"
1786         namespace="##other" processContents="lax"/>
1787     </xs:sequence>
1788     <xs:anyAttribute namespace="##other"/>
1789   </xs:complexType>
1790 </xs:element>
1791
1792 <xs:element name="RelationshipDeletedNotification">
1793   <xs:complexType>
1794     <xs:sequence>
1795       <xs:element ref="muws-p2-xs:Relationship"/>
1796       <xs:any minOccurs="0" maxOccurs="unbounded"
1797         namespace="##other" processContents="lax"/>
1798     </xs:sequence>
1799     <xs:anyAttribute namespace="##other"/>
1800   </xs:complexType>
1801 </xs:element>
1802
1803 <xs:complexType name="RelationshipResourcePropertiesType">
1804   <xs:sequence>
1805     <xs:element ref="muws-p2-xs:Name" minOccurs="0"/>
1806     <xs:element ref="muws-p2-xs:Type"/>
1807     <xs:element ref="muws-p2-xs:Participant"
1808       minOccurs="2" maxOccurs="unbounded"/>
1809   </xs:sequence>
1810 </xs:complexType>
1811
1812 <xs:element name="RelationshipResourceProperties"
1813   type="muws-p2-xs:RelationshipResourcePropertiesType"/>
1814
1815 <xs:element name="QueryRelationshipsByType">
1816   <xs:complexType>
1817     <xs:sequence>
1818       <xs:element name="RequestedType" type="xs:QName"/>
1819     </xs:sequence>
1820   </xs:complexType>
1821 </xs:element>
1822
1823 <xs:element name="QueryRelationshipsByTypeResponse">
1824   <xs:complexType>
1825     <xs:sequence>
1826       <xs:element ref="muws-p2-xs:Relationship"
1827         minOccurs="0" maxOccurs="unbounded"/>
1828     </xs:sequence>
1829   </xs:complexType>
1830 </xs:element>
1831
1832 <xs:element name="CreationNotification">
1833   <xs:complexType>
1834     <xs:sequence>
1835       <xs:element ref="muws-p1-xs:ManageabilityEndpointReference"
1836         minOccurs="0" maxOccurs="unbounded"/>
1837     </xs:sequence>
1838     <xs:anyAttribute namespace="##other"/>
1839   </xs:complexType>
1840 </xs:element>
1841
1842 <xs:element name="DestructionNotification">
1843   <xs:complexType>

```

```

1844     <xs:sequence>
1845         <xs:element ref="muws-p1-xs:ResourceId"
1846             minOccurs="0" />
1847     </xs:sequence>
1848     <xs:anyAttribute namespace="##other" />
1849 </xs:complexType>
1850 </xs:element>
1851
1852 <xs:complexType name="SituationCategoryType">
1853     <xs:complexContent>
1854         <xs:extension base="muws-p2-xs:CategoryType" />
1855     </xs:complexContent>
1856 </xs:complexType>
1857
1858 <xs:complexType name="SubstitutableMsgType">
1859     <xs:sequence>
1860         <xs:element name="Value" type="xs:anySimpleType"
1861             minOccurs="0" maxOccurs="unbounded" />
1862     </xs:sequence>
1863     <xs:attribute name="MsgId" type="xs:string"
1864         use="required" />
1865     <xs:attribute name="MsgIdType" type="xs:anyURI"
1866         use="required" />
1867 </xs:complexType>
1868
1869 <xs:complexType name="SituationType">
1870     <xs:sequence>
1871         <xs:element name="SituationCategory"
1872             type="muws-p2-xs:SituationCategoryType" />
1873         <xs:element name="SuccessDisposition" minOccurs="0">
1874             <xs:simpleType>
1875                 <xs:restriction base="xs:string">
1876                     <xs:enumeration value="Successful" />
1877                     <xs:enumeration value="Unsuccessful" />
1878                 </xs:restriction>
1879             </xs:simpleType>
1880         </xs:element>
1881         <xs:element name="SituationTime" type="xs:dateTime" />
1882         <xs:element name="Priority" type="xs:short"
1883             minOccurs="0" />
1884         <xs:element name="Severity" type="xs:short"
1885             minOccurs="0" />
1886         <xs:element name="Message" type="muws-p2-xs:LangString"
1887             minOccurs="0" />
1888         <xs:element name="SubstitutableMsg"
1889             type="muws-p2-xs:SubstitutableMsgType"
1890             minOccurs="0" />
1891     </xs:sequence>
1892 </xs:complexType>
1893
1894 <xs:element name="Situation" type="muws-p2-xs:SituationType" />
1895
1896
1897
1898 <!-- ##### Metadata description elements ##### -->
1899
1900 <xs:element name="Capability" type="xs:anyURI" />
1901
1902 <xs:complexType name="DialectableExpressionType">
1903     <xs:sequence>
1904         <xs:any namespace="##other" processContents="lax"
1905             minOccurs="0" maxOccurs="unbounded" />
1906     </xs:sequence>

```

```

1907     <xs:attribute name="Dialect" type="xs:anyURI" use="required"/>
1908     <xs:anyAttribute namespace="##other"/>
1909 </xs:complexType>
1910
1911 <xs:element name="ValidWhile"
1912           type="muws-p2-xs:DialectableExpressionType"/>
1913
1914 <xs:element name="Units" type="xs:string"/>
1915
1916 <xs:element name="ChangeType">
1917   <xs:simpleType>
1918     <xs:restriction base="xs:string">
1919       <xs:enumeration value="Counter"/>
1920       <xs:enumeration value="Gauge"/>
1921       <xs:enumeration value="Unknown"/>
1922     </xs:restriction>
1923   </xs:simpleType>
1924 </xs:element>
1925
1926 <xs:element name="TimeScope">
1927   <xs:simpleType>
1928     <xs:restriction base="xs:string">
1929       <xs:enumeration value="Interval"/>
1930       <xs:enumeration value="PointInTime"/>
1931       <xs:enumeration value="SinceReset"/>
1932     </xs:restriction>
1933   </xs:simpleType>
1934 </xs:element>
1935
1936 <xs:element name="GatheringTime">
1937   <xs:simpleType>
1938     <xs:restriction base="xs:string">
1939       <xs:enumeration value="OnChange"/>
1940       <xs:enumeration value="Periodic"/>
1941       <xs:enumeration value="OnDemand"/>
1942       <xs:enumeration value="Unknown"/>
1943     </xs:restriction>
1944   </xs:simpleType>
1945 </xs:element>
1946
1947 <xs:element name="CalculationInterval" type="xs:duration"/>
1948
1949 <xs:element name="MetricGroup" type="xs:anyURI"/>
1950
1951 <xs:element name="PostCondition"
1952           type="muws-p2-xs:DialectableExpressionType"/>
1953
1954 </xs:schema>

```

---

## Appendix D. WSDL elements

1955

```
1956 <?xml version="1.0" encoding="utf-8"?>
1957 <definitions
1958     targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-muws-
1959     part2-1.0.wsdl"
1960     xmlns:muws-p2-wsdl="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-
1961     muws-part2-1.0.wsdl"
1962     xmlns:muws-p2-xs="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-
1963     muws-part2-1.0.xsd"
1964     xmlns:muws-p1-xs="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-
1965     muws-part1-1.0.xsd"
1966     xmlns:wsrf-rp="http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-
1967     ResourceProperties-1.2-draft-01.xsd"
1968     xmlns:xs="http://www.w3.org/2001/XMLSchema"
1969     xmlns="http://schemas.xmlsoap.org/wsdl/">
1970
1971     <types>
1972         <xs:schema elementFormDefault="qualified"
1973             targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-
1974             muws-part2-1.0.wsdl">
1975
1976             <xs:import namespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-
1977             wsdm-muws-part2-1.0.xsd"
1978                 schemaLocation="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-
1979             muws-part2-1.0.xsd"/>
1980
1981             <xs:import namespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-
1982             wsdm-muws-part1-1.0.xsd"
1983                 schemaLocation="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-
1984             muws-part1-1.0.xsd"/>
1985
1986         </xs:schema>
1987     </types>
1988
1989     <message name="QueryRelationshipsByTypeRequest">
1990         <part name="body" element="muws-p2-xs:QueryRelationshipsByType" />
1991     </message>
1992
1993     <message name="QueryRelationshipsByTypeResponse">
1994         <part name="body" element="muws-p2-xs:QueryRelationshipsByTypeResponse" />
1995     </message>
1996
1997
1998
1999     <portType name="Identity"
2000         wsrf-rp:ResourceProperties="muws-p1-xs:IdentityProperties">
2001     </portType>
2002
2003     <portType name="ManageabilityCharacteristics"
2004         wsrf-rp:ResourceProperties="muws-p1-
2005     xs:ManageabilityCharacteristicsProperties">
2006     </portType>
2007
2008     <portType name="CorrelatableProperties"
2009         wsrf-rp:ResourceProperties="muws-p1-xs:CorrelatablePropertiesProperties">
2010     </portType>
2011
2012     <portType name="Description"
2013         wsrf-rp:ResourceProperties="muws-p2-xs:DescriptionProperties">
2014     </portType>
```

```
2015
2016 <portType name="OperationalStatus"
2017     wsrf-rp:ResourceProperties="muws-p2-xs:OperationalStatusProperties">
2018 </portType>
2019
2020 <portType name="Metrics"
2021     wsrf-rp:ResourceProperties="muws-p2-xs:MetricsProperties">
2022 </portType>
2023
2024 <portType name="Relationships"
2025     wsrf-rp:ResourceProperties="muws-p2-xs:RelationshipsProperties">
2026
2027     <operation name="QueryRelationshipsByType">
2028         <input name="QueryRelationshipsByTypeRequest"
2029             message="muws-p2-wsdl:QueryRelationshipsByTypeRequest" />
2030         <output name="QueryRelationshipsByTypeResponse"
2031             message="muws-p2-wsdl:QueryRelationshipsByTypeResponse" />
2032     </operation>
2033
2034 </portType>
2035
2036 <portType name="RelationshipResource"
2037     wsrf-rp:ResourceProperties="muws-p2-xs:RelationshipResourceProperties">
2038 </portType>
2039
2040 </definitions>
```

2041

## Appendix E. Topics

```
2042 <wstop:TopicSpace name="MuwsNotificationTopics"
2043     targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-muws-
2044     events-1.0.xml"
2045     xmlns:wstop="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-
2046     draft-01.xsd"
2047     xmlns:muws-pl-xs="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-
2048     muws-part1-1.0.xsd"
2049     xmlns:muws-p2-xs="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-
2050     muws-part2-1.0.xsd"
2051     xmlns:wsrp="http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-
2052     ResourceProperties-1.2-draft-01.xsd">
2053
2054     <wstop:Topic name="IdentityCapability"
2055         messageTypes="muws-pl-xs:ManagementEvent">
2056     </wstop:Topic>
2057
2058     <wstop:Topic name="ManageabilityCharacteristicsCapability"
2059         messageTypes="muws-pl-xs:ManagementEvent">
2060     </wstop:Topic>
2061
2062     <wstop:Topic name="CorrelatablePropertiesCapability"
2063         messageTypes="muws-pl-xs:ManagementEvent">
2064     </wstop:Topic>
2065
2066     <wstop:Topic name="DescriptionCapability"
2067         messageTypes="muws-pl-xs:ManagementEvent">
2068     </wstop:Topic>
2069
2070     <wstop:Topic name="StateCapability"
2071         messageTypes="muws-pl-xs:ManagementEvent">
2072     </wstop:Topic>
2073
2074     <wstop:Topic name="OperationalStatusCapability"
2075         messageTypes="muws-pl-xs:ManagementEvent">
2076     </wstop:Topic>
2077
2078     <wstop:Topic name="MetricsCapability"
2079         messageTypes="muws-pl-xs:ManagementEvent">
2080     </wstop:Topic>
2081
2082     <wstop:Topic name="ConfigurationCapability"
2083         messageTypes="muws-pl-xs:ManagementEvent">
2084     </wstop:Topic>
2085
2086     <wstop:Topic name="RelationshipsCapability"
2087         messageTypes="muws-pl-xs:ManagementEvent">
2088
2089         <wstop:Topic name="RelationshipCreated"
2090             messageTypes="muws-pl-xs:ManagementEvent">
2091             <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-
2092             19991116">
2093                 //muws-pl-xs:ManagementEvent[ count (muws-p2-
2094                 xs:RelationshipCreatedNotification)=1 ]
2095             </wstop:MessagePattern>
2096         </wstop:Topic>
2097
2098         <wstop:Topic name="RelationshipDeleted"
2099             messageTypes="muws-pl-xs:ManagementEvent">
```

```

2100     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-
2101 19991116">
2102         //muws-p1-xs:ManagementEvent[count(muws-p2-
2103 xs:RelationshipDeletedNotification)=1]
2104     </wstop:MessagePattern>
2105     </wstop:Topic>
2106
2107 </wstop:Topic>
2108
2109 <wstop:Topic name="RelationshipAccessCapability"
2110     messageTypes="muws-p1-xs:ManagementEvent">
2111 </wstop:Topic>
2112
2113 <wstop:Topic name="RelationshipResourceCapability"
2114     messageTypes="muws-p1-xs:ManagementEvent">
2115 </wstop:Topic>
2116
2117 <wstop:Topic name="ManageabilityEndpointCreation"
2118     messageTypes="muws-p1-xs:ManagementEvent">
2119     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-
2120 19991116">
2121         //muws-p1-xs:ManagementEvent[count(muws-p2-xs:CreationNotification)=1]
2122     </wstop:MessagePattern>
2123
2124     <wstop:Topic name="ManageableResourceCreation"
2125         messageTypes="muws-p1-xs:ManagementEvent">
2126         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-
2127 19991116">
2128             //muws-p1-xs:ManagementEvent[count(muws-p2-xs:CreationNotification)=1]
2129         </wstop:MessagePattern>
2130     </wstop:Topic>
2131
2132 </wstop:Topic>
2133
2134 <wstop:Topic name="ManageabilityEndpointDestruction"
2135     messageTypes="muws-p1-xs:ManagementEvent">
2136     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-
2137 19991116">
2138         //muws-p1-xs:ManagementEvent[count(muws-p2-xs:DestructionNotification)=1]
2139     </wstop:MessagePattern>
2140
2141     <wstop:Topic name="ManageableResourceDestruction"
2142         messageTypes="muws-p1-xs:ManagementEvent">
2143         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-
2144 19991116">
2145             //muws-p1-xs:ManagementEvent[count(muws-p2-
2146 xs:DestructionNotification)=1]
2147         </wstop:MessagePattern>
2148     </wstop:Topic>
2149
2150 </wstop:Topic>
2151
2152 </wstop:TopicSpace>

```



---

## 2153 Appendix F. Description of situation types

2154 This appendix defines in more details the situation types introduced in section 2.5.1.

### 2155 AvailabilitySituation

2156 This category deals with the situations reported from the component, regarding its operational  
2157 state and availability. This situation provides a context for operations that can be performed by  
2158 the component to establish if a product is installed, operational and ready to process functional  
2159 requests, or operational and ready or not ready to process management requests. Existing  
2160 message include words like “now ready to take requests”, “online”, and “offline”, for example::

- 2161 • “SOAP connector available at port 8888”

2162

### 2163 CapabilitySituation

2164 This category is specified when a change in capability of a resource occurs. For example, a  
2165 printer has an envelope tray attached to it so that the printer is now has additional paper choices.  
2166 The same category would be used if the envelope tray is removed from the printer.

2167

### 2168 ConfigurationSituation

2169 This category deals with the components identifying configuration changes. Any changes that a  
2170 component makes to its configuration should be logged using this category. Existing message  
2171 include words like “port number is”, “address is”, and “process id”, for example:

- 2172 • “File transfer configured with host='9.27.11.13', port='9090', securityEnabled='false'”

2173

### 2174 StopSituation

2175 This category deals with the shutdown process for a component. Messages that indicate that a  
2176 component has begun to stop, that it has stopped, or that the stopping process has failed all fall  
2177 into this category. Existing messages include words like “stop”, “stopping”, “stopped”,  
2178 “completed”, and “exiting”, for example:

- 2179 • “Application stopped: myApp.exe”
- 2180 • “An error occurred while stopping myApp.exe”
- 2181 • “Stopping the JMS provider”

2182

### 2183 StartSituation

2184 This category deals with the startup process for a component. Messages that indicate that a  
2185 component has begun the startup process, that it has finished the startup process, or that it has  
2186 aborted the startup process all fall into this category. Existing messages include words like  
2187 “starting”, “started”, “initializing”, and “initialized”, for example:

- 2188 • “XYZ protocol support was successfully started”
- 2189 • “XYZ protocol support failed to start”
- 2190 • “Starting EJB: myEjb.jar”

2191

### 2192 RequestSituation

2193 This category is used in situations that a component uses to identify the completion status of a  
2194 request. Typically, these requests are complex management tasks or transactions that a  
2195 component undertakes on behalf of a requestor and not the mainline simple requests or  
2196 transactions. Existing messages are of the form “request started” or “request completed” as in  
2197 phrases like “configuration synchronization started”, and “backup procedure complete”, for  
2198 example:

2199       • “Configuration synchronization completed”  
2200 Note that events generated from requests that start up or stop a resource would be categorized  
2201 as **StartSituation** or **StopSituation** respectively because they are higher precedent than  
2202 **RequestSituation**.

2203

### 2204 **DestroySituation**

2205 This category deals with the situations occurring when an entity or component was removed or  
2206 destroyed. Messages telling that a document was destroyed or a file was deleted all fall into this  
2207 category. Existing messages include phrases like “was destroyed”, “about to remove”, and “no  
2208 longer exists”, for example:

2209       • “The connection pool was destroyed for data source foo”

2210

### 2211 **CreateSituation**

2212 This category deals with the situations occurring when a component creates an entity. Messages  
2213 telling that a document was created, or a file was created, or an Enterprise JavaBean (EJB) was  
2214 created all fall into this category. Existing message include words like was created, about to  
2215 create, and now exists, for example:

2216       • “New log file was created”

2217

### 2218 **DependencySituation**

2219 This category deals with the situations where components cannot find some component or  
2220 feature that they require. This category includes messages about not finding the “version” of the  
2221 component that was expected. Messages that say a resource was not found, or that an  
2222 application or subsystem that was unavailable, also fall into this category. Existing messages  
2223 include words like “could not find”, and “no such component”, for example:

2224       • “Error encountered while deploying database schema: no database found”

2225

2226

### 2227 **ConnectSituation**

2228 This category deals with the situations related to aspects about a connection attempt from one  
2229 component to another component. Messages that say a connection failed, that a connection was  
2230 created, or that a connection was ended all fall into this category. Existing messages include  
2231 words like “connection reset”, “connection failed”, and “failed to get a connection”, for example:

2232       • “Connection creation failed”

2233       • “Connection with http://foo.com created”

2234       • “Failed to close a connection”

2235

### 2236 **ReportSituation**

2237 This category deals with situations that occur as a result of some setting or occurrence that  
2238 causes the resource to asynchronously report various types of data. Types of information that  
2239 falls into this category are:

2240

2241       • **Exception related** – some exception has occurred within the resource and it not covered  
2242 by any other category.

2243       • **Performance related** – some event occurs, that does not fall into any other category,  
2244 that has affected performance in some way. For example, weather conditions may be  
2245 affected line quality and network speeds are affected.

- 2246 • **Security related** – some security issue has been detected, like the cabinet door to a
- 2247 secure piece of equipment has been opened or an attack of some sort has been
- 2248 detected.
- 2249 • **Heartbeat related** – the resource has been configured to periodically report a ‘heartbeat’.
- 2250 • **Status related** – some change of status that does not affect availability or capability of
- 2251 the resource has been detected. For example, printer ink cartridge is low.
- 2252 • **Log related** – the resource has been configured to generate a log entry based on some
- 2253 event or at a fixed interval. This category identifies this event as a requested log entry.
- 2254 • **Debug related** – the resource has been enabled to turn on diagnostic information flow
- 2255 and will report the information within this category.
- 2256 • **Trace related** – the resource has been enabled to run trace information and reports this
- 2257 information using this category

2258

#### 2259 **OtherSituation**

2260

2261 This category is for those events that do not fall into any other category. Note that this category  
2262 is defined for syntactic completeness but any events placed in this category will not be able to be  
2263 effectively correlated and its use is therefore discouraged unless absolutely necessary.

2264

2265