



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

OASIS Standard, March 8th 2005

Document identifier:

wsdm-muws-part2-1.0

Location:

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

Editor:

William Vambenepe, Hewlett-Packard <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 an OASIS standard.

Committee members should send comments on this specification to the wsdm@lists.oasis-open.org list. Others should subscribe and send comments to the wsdm-comment@lists.oasis-open.org list. To subscribe, send an email message to 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/wsdm-muws-part2-1.0-errata.pdf>

36 **Table of Contents**

37	1	Introduction.....	4
38	2	Use of the Web Services Platform.....	5
39	2.1	Use of WS-Addressing and the WS-Resource concept.....	5
40	2.2	Use of WS-Resource Properties.....	5
41	2.3	Use of WS-Notification.....	6
42	2.4	Metadata.....	6
43	2.4.1	Metadata applicable to all aspects of manageability interfaces.....	6
44	2.4.2	Metadata applicable to properties.....	7
45	2.4.3	Operations.....	8
46	2.5	Events.....	8
47	2.5.1	Event Format.....	8
48	2.5.2	Topics for capabilities.....	11
49	2.6	Representation of Categorization Taxonomies in XML.....	11
50	3	Capabilities applicable to manageable resources.....	13
51	3.1	Description.....	13
52	3.1.1	Definition.....	13
53	3.1.2	Properties.....	13
54	3.1.3	Events.....	14
55	3.2	State.....	14
56	3.2.1	Definition.....	14
57	3.2.2	Describing State Models.....	15
58	3.2.3	Information Markup Declarations.....	16
59	3.2.4	Properties.....	17
60	3.2.5	Operations.....	18
61	3.2.6	Events.....	18
62	3.3	Operational Status.....	19
63	3.3.1	Definition.....	19
64	3.3.2	Properties.....	19
65	3.3.3	Events.....	20
66	3.4	Metrics.....	20
67	3.4.1	Definition.....	20
68	3.4.2	Information Markup Declarations.....	20
69	3.4.3	Metadata.....	21
70	3.4.4	Properties.....	23
71	3.4.5	Events.....	23
72	3.5	Configuration.....	23
73	3.5.1	Definition.....	23
74	3.5.2	Properties.....	23
75	3.5.3	Operations.....	24
76	3.5.4	Events.....	24
77	4	Capabilities applicable to management in general.....	25

78	4.1	Relationships.....	25
79	4.1.1	Definition.....	25
80	4.1.2	Information Markup Declarations	26
81	4.1.3	Properties	28
82	4.1.4	Operations	29
83	4.1.5	Events.....	30
84	4.2	Relationship Access Capability.....	30
85	4.2.1	Definition.....	30
86	4.2.2	Events.....	31
87	4.3	Relationship Resource Capability	31
88	4.3.1	Definition.....	31
89	4.3.2	Properties	31
90	4.3.3	Events.....	32
91	4.4	Advertisement	32
92	4.4.1	Definition.....	32
93	4.4.2	Events.....	33
94	5	Discovery.....	35
95	5.1	Discovery using Relationships	35
96	5.2	Discovery using Registries.....	35
97	6	References.....	37
98	6.1	Normative.....	37
99	6.2	Non-normative.....	38
100	Appendix A.	Acknowledgements.....	39
101	Appendix B.	Notices.....	40
102	Appendix C.	Schemas.....	41
103	Appendix D.	WSDL elements	47
104	Appendix E.	Topics.....	49
105	Appendix F.	Description of situation types.....	51
106			

107 **1 Introduction**

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

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

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

117 The following namespaces are used, unless specified otherwise.

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

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

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

2 Use of the Web Services Platform

122

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

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

127

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

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

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

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

2.2 Use of WS-Resource Properties

154

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

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

168 2.3 Use of WS-Notification

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

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

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

184 2.4 Metadata

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

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

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

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

203 2.4.1 Metadata applicable to all aspects of manageability interfaces

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

206

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

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

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

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

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

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

232
233

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

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

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

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

247 2.4.2 Metadata applicable to properties

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

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

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

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

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

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

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

266 2.4.3 Operations

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

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

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

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

276

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

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

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

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

290 2.5 Events

291 2.5.1 Event Format

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

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

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

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

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

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

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

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

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

- 351 • Successful
 - 352 • Unsuccessful
- 353

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

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

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

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

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

- 373 • 6 (Fatal): a condition is unrecoverable and the service is no longer available.
- 374 • 5 (Critical): a condition affecting the service has occurred. Immediate corrective action is
375 required.
- 376 • 4 (Major): a problem of relatively high severity has occurred. It is likely that normal use of
377 the service is impeded.
- 378 • 3 (Minor): a problem of relatively low severity has occurred. It is unlikely that normal use
379 of the service is impeded.
- 380 • 2 (Warning): a problem affecting the service may occur. Diagnostic and corrective action
381 is recommended.
- 382 • 1 (Information): a message output considered as normal and expected. For example, a
383 process begins, a process finishes, or status information is displayed.
- 384 • 0 (Unknown): a severity level cannot be determined.

385

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

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

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

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

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

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

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

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

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

430 2.5.2 Topics for capabilities

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

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

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

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

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

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

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

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

456 2.6 Representation of Categorization Taxonomies in XML

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

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

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

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

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

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

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

480 Category information MUST be declared as follows:

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

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

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

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

505 3 Capabilities applicable to manageable 506 resources

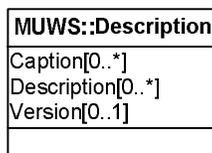
507 This section defines capabilities applicable to manageable resources. The capabilities defined in
508 this section complement the capabilities defined in MUWS Part 1.

509 3.1 Description

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

512 3.1.1 Definition

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



514
515

Figure 1: MUWS Description

516 3.1.2 Properties

517 This capability defines the following properties:

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

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

525 Metadata for *Caption*:

526 It is *Mutable*

527 It is *Modifiable*

528 It has the following *Capability* metadata item:

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

532

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

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

540 Metadata for *Description*:

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

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

547

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

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

553 Metadata for *Version*:

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

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

560 3.1.3 Events

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

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

566 3.2 State

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

569 3.2.1 Definition

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

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

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

582 **3.2.2 Describing State Models**

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

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

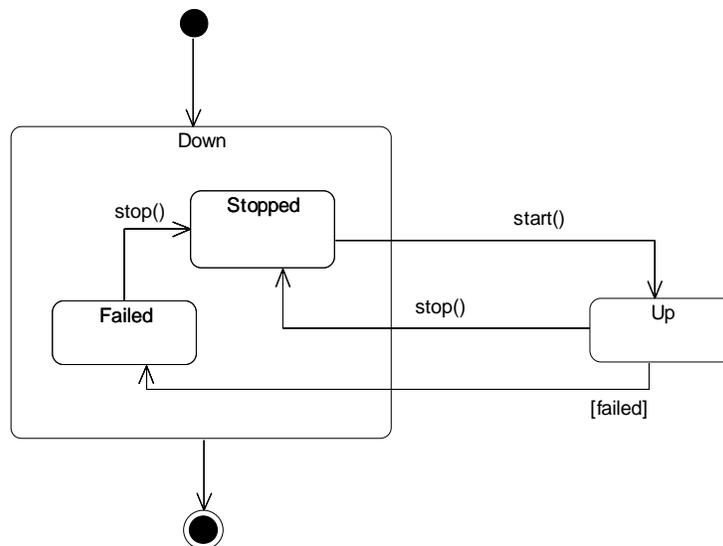
588 States can have sub-states that MUST be wholly contained within a higher-level state.

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

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

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

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



600

601

Figure 2: Example Operational State Model

602 In this example, the state machine is identified by URI
603 <http://example.com/StateModels/SimpleOperationalState>, bound to namespace prefix *exns*.

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

- 606 • *exns:Down*

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

- 609 • *exns:Stopped*

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

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

623 3.2.3 Information Markup Declarations

624 3.2.3.1 Representation of States

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

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

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

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

639 A state MUST be declared as follows:

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

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

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

656 3.2.3.2 Representation of state

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

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

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

665 3.2.3.3 Representation of state transition

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

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

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

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

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

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

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

687 3.2.4 Properties

688 This capability does not define any standard property.

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

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

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

701 3.2.4.1 Example

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

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

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

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

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

732 3.2.5 Operations

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

736 3.2.6 Events

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

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

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

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

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

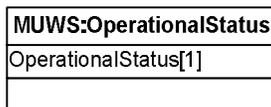
755 3.3 Operational Status

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

758 3.3.1 Definition

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

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



768
 769

Figure 3: Operational Status

770 3.3.2 Properties

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

```

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

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

```

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

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

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

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

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

801 3.3.3 Events

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

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

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

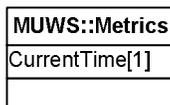
809 3.4 Metrics

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

812 3.4.1 Definition

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

817 Figure 4 presents the *Metrics* capability.



818
819

Figure 4: MUWS metrics

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

828 3.4.2 Information Markup Declarations

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

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

835 </xs:attributeGroup>

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

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

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

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

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

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

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

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

866 3.4.3 Metadata

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

868 It is *Mutable*

869 It is not *Modifiable*

870 It has the following *Capability* metadata item:

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

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

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

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

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

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

887

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

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

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

908

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

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

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

922

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

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

930

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

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

938 3.4.4 Properties

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

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

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

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

951 3.4.5 Events

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

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

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

960 3.5 Configuration

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

963 3.5.1 Definition

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

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

968 3.5.2 Properties

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

972 It is *Mutable*

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

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

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

980 **3.5.3 Operations**

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

983 **3.5.4 Events**

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

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

4 Capabilities applicable to management in general

989

990

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

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

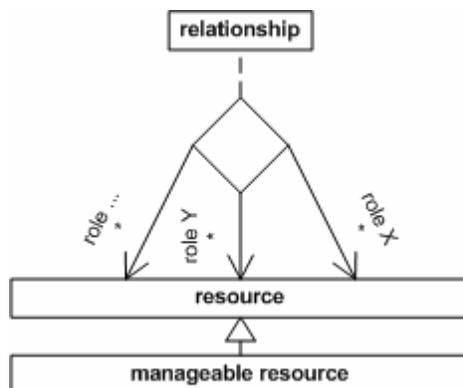
999 The function of a management-related capability is related to the management of a resource, but
1000 it is not necessarily offered directly by a manageability endpoint of a resource. For example, the
1001 capability to help a manageability consumer discover a new manageable resource can be
1002 provided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

4.1 Relationships

1008 The manageability capability URI for this capability is
1009 <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Relationships>

4.1.1 Definition

1011 A relationship is an N-ary association between resources. A relationship may have properties and
1012 other characteristics. One of these properties is a type that conveys the semantic of the
1013 relationship. The resources involved in the relationship are called participants. Each participant
1014 has a role in the relationship. The participants may or may not be manageable resources in the
1015 MUWS sense. The notion of "direction" of a relationship is a semantic interpretation based on role
1016 definitions. There could be many instances of relationships between many instances of
1017 resources. The arrows in Figure 5 depict navigability, which means that by following the arrow
1018 one could resolve what the end points to (reference).

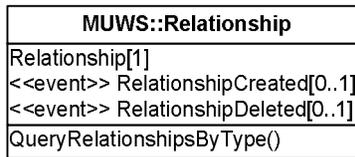


1019

1020

Figure 5: Relationship conceptual model

1021 Note that this capability is not limited to manageable resources and can be exposed by any
 1022 resource that wants to expose relationships that it knows about.
 1023 Figure 6 is a UML representation of the relationship capability.



1024
 1025

Figure 6: Relationship capability

1026 A relationships may become stale. The information about a relationship should be validated,
 1027 either manually or automatically, before it can be relied upon. Exposing the information about a
 1028 relationship should be considered a potential security risk if a participating resource should not be
 1029 visible for security reasons.

1030 4.1.2 Information Markup Declarations

1031 4.1.2.1 Representation of Categories of Relationships

1032 A relationship may be categorized as a certain type of relationship. A relationship type defines the
 1033 semantics of the relationship. One relationship type may be a specialization or generalization of
 1034 another type..This defines a taxonomy of relationship categories. MUWS defines a way to
 1035 represent a type and its taxonomy lineage, but the actual definition of a relationship type is
 1036 specific to a resource management model. Therefore, no relationship type is defined by MUWS.
 1037 In other words, MUWS specifies only the mechanism to convey a relationship type, or category,
 1038 in XML as follows.

1039 *RelationshipTypeType* type is declared as follows

```
1040 <xs:complexType name="RelationshipTypeType">
1041   <xs:complexContent>
1042     <xs:extension base="muws-p2-xs:CategoryType" />
1043   </xs:complexContent>
1044 </xs:complexType>
```

1045 The *RelationshipTypeType* type is used to declare an XML element containing instances of
 1046 relationship type information.

1047 The relationship type information MUST be declared as follows:

- 1048 • An XML element declaring which QName identifies the semantics of a relationship type..
- 1049 • The XML element MUST be declared with an XML Schema type that is a restriction of
 1050 *RelationshipTypeType*.
- 1051 • The contents of the XML element MUST be either
 - 1052 • The only one XML element corresponding to the generalization of the currently
 1053 declared relationship type
 - 1054 • The empty sequence, if the currently declared relationship type does not have a
 1055 generalization, such as the top of a taxonomy.

1056 For example, the "USB attached" relationship type may be generalized to the "Bus connected"
 1057 type which, in turn, may be generalized to the "Generally linked" type. An instance of the "USB
 1058 attached" relationship type information may be represented in the following XML fragment by
 1059 using the rules described above:

```
1060 <my:RelationshipTypeInstanceElement xsi:type="RelationshipTypeType">
1061   <usb:Attached>
1062     <bus:Connected>
1063     <generally:Linked/>
```

1064
1065
1066

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

1067 4.1.2.2 Representation of an Instance of a Relationship

1068 MUWS defines the following Global Element Declaration (GED) to represent an instance of a
1069 relationship.

1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084

```
<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>
```

1085 **muws-p2-xs:Relationship/muws-p2-xs:Name** is a human readable name for a relationship.
1086 *Name* should not be used for machine reasoning about the semantics of a relationship. Type
1087 should be used instead. This element is OPTIONAL.

1088 **muws-p2-xs:Relationship/muws-p2-xs:Type** is the relationship type this relationship belongs
1089 to. Examples of such types include linkage, containment, or dependency. MUWS does not define
1090 any specific relationship type. This is left to domain-specific models. MUWS only defines a way to
1091 convey the type as part of the representation of a relationship. In order to allow relationships to be
1092 defined as part of a taxonomy, the mechanism used by MUWS to represent relationship types
1093 leverages the *muws-p2-xs:CategoryType* type defined in section 2.6. This element is
1094 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:HostName>  
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

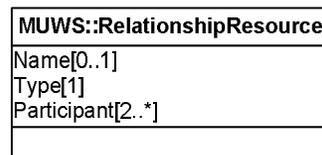
1279 The manageability capability URI for this capability is
 1280 <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/RelationshipResource>

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:*
1482 *Management using Web Services (MUWS 1.0) Part 1*, OASIS Committee
1483 Draft, December 2004, [http://docs.oasis-open.org/wsdm/2004/12/wsdm-](http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part1-1.0.pdf)
1484 [muws-part1-1.0.pdf](http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part1-1.0.pdf)

1485

1486 **[XML1.0 3rd Edition]**
1487 Tim Bray, et al., *Extensible Markup Language (XML) 1.0 (Third Edition)*,
1488 W3C Recommendation, February 2004, <http://www.w3.org/TR/REC-xml>

1489

1490 **[XML Schema Part 1]**
1491 Henry S. Thompson, et al. *XML Schema Part 1: Structures*, W3C
1492 Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-1/>

1493

1494 **[XML Schema Part 2]**
1495 Paul V. Biron, et al. *XML Schema Part 2: Datatypes*, W3C
1496 Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-2/>

1497

1498 **[WSDL]** Erik Christensen, et al., *Web services Description Language (WSDL)*
1499 *1.1*, W3C Note, March 2001, <http://www.w3.org/TR/wsdl>

1500

1501 **[WS-Resource]** Steve Graham, et al. *Web Service Resource 1.2 (WS-Resource)*, OASIS
1502 Working Draft, October 2004, [http://www.oasis-](http://www.oasis-open.org/apps/org/workgroup/wsrp/download.php/9547/wsrp-WS-Resource-1.2-draft-01.doc)
1503 [open.org/apps/org/workgroup/wsrp/download.php/9547/wsrp-WS-](http://www.oasis-open.org/apps/org/workgroup/wsrp/download.php/9547/wsrp-WS-Resource-1.2-draft-01.doc)
1504 [Resource-1.2-draft-01.doc](http://www.oasis-open.org/apps/org/workgroup/wsrp/download.php/9547/wsrp-WS-Resource-1.2-draft-01.doc)

1505

1506 **[WS-Addressing]** Don Box, et al., *Web services Addressing (WS-Addressing)*, W3C
1507 Member Submission, August 2004,
1508 <http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/>

1509

1510 **[WS-RP]** Steve Graham, et al., *Web Services Resource Properties 1.2 (WS-*
1511 *ResourceProperties)*, OASIS Working Draft, June 2004,
1512 [http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-](http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-1.2-draft-04.pdf)
1513 [1.2-draft-04.pdf](http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-1.2-draft-04.pdf)

1514

1515 **[XPath 1.0]** James Clark, et al., *XML Path Language (XPath) Version 1.0*, W3C
1516 Recommendation, November 1999, [http://www.w3.org/TR/1999/REC-](http://www.w3.org/TR/1999/REC-xpath-19991116)
1517 [xpath-19991116](http://www.w3.org/TR/1999/REC-xpath-19991116)

1518

1519 **[WSN]** Steve Graham, et al., *Web Services Base Notification 1.2 (WS-*
1520 *BaseNotification)*, OASIS Working Draft, June 2004, [http://docs.oasis-](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf)
1521 [open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf)

1522

1523 **[WST]** William Vambenepe, *Web Services Topics 1.2 (WS-Topics)*, OASIS
1524 Working Draft, July 2004, [http://docs.oasis-open.org/wsn/2004/06/wsn-](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.pdf)
1525 [WS-Topics-1.2-draft-01.pdf](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.pdf)

- 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/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-03.pdf)
1533 [open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-03.pdf](http://docs.oasis-open.org/wsrf/2004/06/wsrf-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/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-02.pdf)
1537 [open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-02.pdf](http://docs.oasis-open.org/wsrf/2004/06/wsrf-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/wsrf-WS-](http://www.oasis-open.org/committees/download.php/9758/wsrf-WS-ResourceMetadataDescriptor-1.0-draft-01.PDF)
1549 [ResourceMetadataDescriptor-1.0-draft-01.PDF](http://www.oasis-open.org/committees/download.php/9758/wsrf-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 2003-2005. *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/wsdm-muws-
1599     part2.xsd"
1600     xmlns:muws-p2-xs="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1601     part2.xsd"
1602     xmlns:muws-p1-xs="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1603     part1.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/wsdm-muws-
1609     part1.xsd"
1610             schemaLocation="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-
1611     muws-part1.xsd"/>
1612     <xs:import 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" mixed="true">
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/wsdm-muws-
1959     part2.wsdl"
1960     xmlns:muws-p2-wsdl="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1961     part2.wsdl"
1962     xmlns:muws-p2-xs="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1963     part2.xsd"
1964     xmlns:muws-p1-xs="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1965     part1.xsd"
1966     xmlns:wsrf-rp="http://docs.oasis-open.org/wsrf/2004/06/wsrf-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/wsdm-
1974             muws-part2.wsdl">
1975
1976             <xs:import namespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-
1977             muws-part2.xsd"
1978                 schemaLocation="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-
1979             muws-part2.xsd"/>
1980
1981             <xs:import namespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-
1982             muws-part1.xsd"
1983                 schemaLocation="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-
1984             muws-part1.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/wsdm-muws-
2044     part2-events.xml"
2045     xmlns:muws-p1-xs="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
2046     part1.xsd"
2047     xmlns:muws-p2-xs="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
2048     part2.xsd"
2049     xmlns:wstop="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-
2050     draft-01.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-p1-xs:ManagementEvent" >
2056     </wstop:Topic>
2057
2058     <wstop:Topic name="ManageabilityCharacteristicsCapability"
2059         messageTypes="muws-p1-xs:ManagementEvent" >
2060     </wstop:Topic>
2061
2062     <wstop:Topic name="CorrelatablePropertiesCapability"
2063         messageTypes="muws-p1-xs:ManagementEvent" >
2064     </wstop:Topic>
2065
2066     <wstop:Topic name="DescriptionCapability"
2067         messageTypes="muws-p1-xs:ManagementEvent" >
2068     </wstop:Topic>
2069
2070     <wstop:Topic name="StateCapability"
2071         messageTypes="muws-p1-xs:ManagementEvent" >
2072     </wstop:Topic>
2073
2074     <wstop:Topic name="OperationalStatusCapability"
2075         messageTypes="muws-p1-xs:ManagementEvent" >
2076     </wstop:Topic>
2077
2078     <wstop:Topic name="MetricsCapability"
2079         messageTypes="muws-p1-xs:ManagementEvent" >
2080     </wstop:Topic>
2081
2082     <wstop:Topic name="ConfigurationCapability"
2083         messageTypes="muws-p1-xs:ManagementEvent" >
2084     </wstop:Topic>
2085
2086     <wstop:Topic name="RelationshipsCapability"
2087         messageTypes="muws-p1-xs:ManagementEvent" >
2088
2089         <wstop:Topic name="RelationshipCreated"
2090             messageTypes="muws-p1-xs:ManagementEvent" >
2091             <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-
2092             19991116">
2093                 //muws-p1-xs:ManagementEvent[ count (muws-p2-
2094                 xs:RelationshipCreatedNotification)=1 ]
2095             </wstop:MessagePattern>
2096         </wstop:Topic>
2097
2098     <wstop:Topic name="RelationshipDeleted"
2099         messageTypes="muws-p1-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