



Web Services Distributed Management: Management Using Web Services (MUWS 1.1) Part 2

OASIS Standard, 01 August 2006

Document identifier:

wsdm-muws2-1.1-spec-os-01

Location:

<http://docs.oasis-open.org/wsdm/wsdm-muws2-1.1-spec-os-01.pdf>

Technical Committee:

OASIS Web Services Distributed Management TC

Chair(s):

Heather Kreger, IBM, <kreger@us.ibm.com>

Editor:

*Vaughn Bullard, AmberPoint, Inc. <vbullard@amberpoint.com>
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 was last revised or approved by the membership of OASIS on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at www.oasis-open.org/committees/wsdm/.

39 For information on whether any patents have been disclosed that may be essential to
40 implementing this specification, and any offers of patent licensing terms, please refer to
41 the Intellectual Property Rights section of the Technical Committee web page
42 (www.oasis-open.org/committees/wsdm/ipr.php).
43 The non-normative errata page for this specification is located at [www.oasis-](http://www.oasis-open.org/committees/wsdm)
44 [open.org/committees/wsdm](http://www.oasis-open.org/committees/wsdm).
45

46 Table of Contents

47	1	Introduction	5
48	2	Use of the Web Services Platform	6
49	2.1	Use of WS-Addressing and the WS-Resource concept.....	6
50	2.2	Use of WS-Resource Properties	6
51	2.3	Use of WS-Notification	6
52	2.4	Metadata	7
53	2.4.1	Metadata applicable to all aspects of manageability capabilities	7
54	2.4.2	Metadata applicable to properties.....	8
55	2.4.3	Operations	9
56	2.5	Events	9
57	2.5.1	Event Format	9
58	2.5.2	Topics for capabilities	14
59	2.6	Representation of Categorization Taxonomies in XML.....	14
60	3	Capabilities applicable to manageable resources	16
61	3.1	Description	16
62	3.1.1	Definition	16
63	3.1.2	Properties.....	16
64	3.1.3	Events	17
65	3.2	State	17
66	3.2.1	Definition	17
67	3.2.2	Describing State Models	17
68	3.2.3	Information Markup Declarations	19
69	3.2.4	Properties.....	20
70	3.2.5	Operations	21
71	3.2.6	Events	21
72	3.3	Operational Status.....	21
73	3.3.1	Definition	22
74	3.3.2	Properties.....	22
75	3.3.3	Events	22
76	3.4	Metrics.....	23
77	3.4.1	Definition	23
78	3.4.2	Information Markup Declarations	23
79	3.4.3	Metadata	24
80	3.4.4	Properties.....	26
81	3.4.5	Events	26
82	3.5	Configuration	27
83	3.5.1	Definition	27
84	3.5.2	Properties.....	27
85	3.5.3	Operations	27
86	3.5.4	Events	27
87	4	Capabilities applicable to management in general	28

88	4.1	Relationships.....	28
89	4.1.1	Definition.....	28
90	4.1.2	Information Markup Declarations.....	28
91	4.1.3	Properties.....	31
92	4.1.4	Operations.....	31
93	4.1.5	Events.....	32
94	4.2	Relationship Resource Capability.....	33
95	4.2.1	Definition.....	33
96	4.2.2	Properties.....	33
97	4.2.3	Events.....	34
98	4.3	Advertisement.....	34
99	4.3.1	Definition.....	34
100	4.3.2	Events.....	34
101	5	Discovery.....	37
102	5.1	Discovery using Relationships.....	37
103	5.2	Discovery using Registries.....	37
104	6	References.....	39
105	6.1	Normative.....	39
106	6.2	Non-normative.....	40
107	Appendix A.	Acknowledgements.....	41
108	Appendix B.	Notices.....	42
109	Appendix C.	Schemas.....	43
110	Appendix D.	WSDL elements.....	54
111	Appendix E.	Topics.....	55
112	Appendix F.	Description of situation types.....	57
113			

114 1 Introduction

115 This document, MUWS Part 2, builds upon the foundation provided by **[MUWS Part 1]**. All of the
116 normative text presented in MUWS Part 1 is considered normative text for MUWS Part 2. All
117 informational text presented in MUWS Part 1 is relevant informational text for MUWS Part 2.
118 Compliance with MUWS Part 1 is REQUIRED for every aspect of MUWS Part 2.

119 The text of this specification along with Appendix C (Schemas), Appendix D (WSDL Elements),
120 Appendix E (Topics) and Appendix F (Description of Situation Types) is considered normative
121 with the following exceptions: the abstract, the examples and any section explicitly marked as
122 non-normative.

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

124 The following namespaces are used, unless specified otherwise.

Prefix	Namespace
muws1	http://docs.oasis-open.org/wsdm/muws1-2.xsd
muws2	http://docs.oasis-open.org/wsdm/muws2-2.xsd
muwsw	http://docs.oasis-open.org/wsdm/muws-2.wsdl
muwse	http://docs.oasis-open.org/wsdm/muwse-2.xml
wsnt	http://docs.oasis-open.org/wsn/b-2.xsd
wstop	http://docs.oasis-open.org/wsn/t-1
wsrf-rp	http://docs.oasis-open.org/wsrf/rp-2
wssg	http://docs.oasis-open.org/wsrf/sgw-2.wsdl
wsdl	http://www.w3.org/2002/07/wsdl
wsa	http://www.w3.org/TR/ws-addr-core
soap	http://schemas.xmlsoap.org/soap/envelope/ or http://www.w3.org/2002/12/soap-envelope
xs	http://www.w3.org/2001/XMLSchema

125 XML elements (**[XML1.0 3rd Edition]**) and schema (**[XML Schema Part 1]** and **[XML Schema**
126 **Part 2]**) types introduced in this section belong to the namespace mapped to “muws2”.

127 WSDL (**[WSDL]**) elements introduced in this section belong to the namespace mapped to “muws-
128 2.wsdl”.

2 Use of the Web Services Platform

129

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

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

134
135 MUWS Part 2 depends upon concepts presented in the Web Services Resources Framework
136 (WSRF). A manageable resource is a refinement of a WSRF resource. A WS-Resource is
137 defined as the actual composition of a resource and a web service from which the resource can
138 be accessed. In addition, a reference to a manageability endpoint relies upon reference
139 mechanisms as defined in the WSRF WS-Resource Specification **[WS-Resource]**, and more
140 specifically, leverages and refines the endpoint reference concept, as defined in WS-Addressing.

141 The MUWS specification does not currently define how to obtain an EPR. Currently, to obtain an
142 EPR, there may be some out-of-band agreement between a service provider and a manageability
143 consumer. Possibly, some future version of the MUWS specification might clarify and standardize
144 an approach to obtain an EPR. This specification provides some guidelines on discovering EPRs
145 for manageability endpoints and the WSDM Primers explain how to enable and use discovery.

146 In the specific case where a manageability endpoint corresponds to one and only one
147 manageable resource, then either the endpoint reference concept defined in WS-Addressing can
148 be used. If no endpoint reference exists, then the consumer can use a URL to identify the
149 manageable resource. Following this reasoning, a manageability consumer without an endpoint
150 reference for a manageable resource MAY try to invoke manageability operations without
151 including an endpoint reference. If such an invocation succeeds, the manageability consumer
152 can infer it is accessing a manageable resource through a manageability provider.

2.2 Use of WS-Resource Properties

153
154 Management properties as defined in MUWS are represented as WSRF properties, and use the
155 mechanisms defined in WS-ResourceProperties (**[WS-RP]**). In other words, each manageable
156 resource exposes a resource properties document containing, as children of the document root,
157 all the properties of the manageable resource. The manageable resource then makes this
158 document available, as described in WS-ResourceProperties.

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

2.3 Use of WS-Notification

167
168 MUWS uses the notification mechanism described by WS-BaseNotification (**[WSN]**). If a
169 manageability capability includes an ability to offer events to a consumer, then the definition of
170 the capability SHALL include topic space, as described in WS-Topics (**[WST]**). The topic space
171 MUST contain an appropriate set of topics for the events offered by the capability. As described
172 in MUWS Part 1, an event is defined by a topic QName and a content element. The topic is
173 mapped to the topic of the event, as defined by **[WST]**.

174 As specified by WS-BaseNotification, whether the event payload (of type
175 *muws1:ManagementEvent*) is the first child of the SOAP **[MOWS]**
176 Kirk Wilson, *Web Services Distributed Management:
177 Management of Web Services (WSDM-MOWS) 1.1*, OASIS Committee
178 Draft, February 2006, [http://docs.oasis-open.org/wsdm/wsdm-mows-1.1-](http://docs.oasis-open.org/wsdm/wsdm-mows-1.1-spec-os-01.pdf)
179 [spec-os-01.pdf](http://docs.oasis-open.org/wsdm/wsdm-mows-1.1-spec-os-01.pdf)

180

181 **[SOAP]**) body or whether it is wrapped in a *wsnt:Notify* element is determined based on whether
182 the *wsnt:UseNotify* element in the subscription message is set to *true* or *false*.

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

188 2.4 Metadata

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

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

202 This MUWS specification does not provide any description of how metadata is associated with a
203 type of manageable resource, stored, or made available.

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

207 2.4.1 Metadata applicable to all aspects of manageability capabilities

208 MUWS defines metadata elements applicable to all aspects of a manageability capability
209 (operations, properties, events...). These elements are:

210 `<muws2:Capability>xs:anyURI</muws2:Capability> *`

211 **muws2:Capability** metadata element SHOULD be provided for any MUWS aspect of a
212 manageability interface. This enables discovery of aspects of an interface associated with a
213 capability. This URI element identifies which capability an aspect is associated with.

214

215 This metadata element indicates the classification of an aspect of an interface according to an
216 intended capability, or capabilities. For example, an aspect may be classified as a metric, or, as
217 a configuration property. A property may be relevant to more than one capability. For example, a
218 configuration property of a computer system contains the IP address but this same property could
219 also be used for identification purposes.

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

- 223 • <http://docs.oasis-open.org/wsdm/muws/capabilities/Identity>
224 Identity capability. See [MUWS Part 1].
- 225 • <http://docs.oasis-open.org/wsdm/muws/capabilities/Configuration>
226 Configuration property. See section 3.5.
- 227 • <http://docs.oasis-open.org/wsdm/muws/capabilities/CorrelatableProperties>
228 "Correlatable Properties" capability. See [MUWS Part 1].
- 229 • <http://docs.oasis-open.org/wsdm/muws/capabilities/State>
230 State capability. See section 3.1.3.
- 231 • <http://docs.oasis-open.org/wsdm/muws/capabilities/Metrics>
232 Metrics capability. See section 3.4.
- 233 • *User defined*
234 A user defined capability that extends, or, is different from, a standard capability defined
235 in MUWS.
236
237

```
238 <muws2:ValidWhile Dialect="xs:anyURI" > {any} * </muws2:ValidWhile>
```

239 **muws2:ValidWhile** contains a statement that, when true, asserts that the interface aspect to
240 which this metadata element is related is valid. This is used, for example, to express the fact that
241 an operation can only be invoked when certain properties have certain values.

242 **muws2:ValidWhile/@muws2:Dialect** identifies how the statement in *muws2:ValidWhile* is built
243 and what rules govern its evaluation. MUWS defines one possible value for this element. Other
244 values can also be defined.

245 The value defined by MUWS is <http://www.w3.org/TR/1999/REC-xpath-19991116>. When this
246 dialect is used, the content of *muws2:ValidWhile* is an [XPath 1.0] expression. This expression is
247 evaluated against the resource properties document of the manageable resource. If the XPath
248 expression evaluates to a Boolean value of *true*, or if it evaluates to a non-empty non-boolean
249 value without any errors, then the statement is considered true.

250 2.4.2 Metadata applicable to properties

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

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

261 Schema to represent general purpose metadata should be composed from a metadata
262 specification, for example, the WS-Resource Metadata Descriptor [WSRMD], as developed in the
263 WS-RF OASIS technical committee. (***Special note*: Although the WSRF Technical committee is
264 currently working on the WSRMD, the WSDM committee will probably reference the WSRMD in a
265 future release of the WSDM specification.)

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

```
268 <muws2:Units>xs:string</muws2:Units>
```

269 **muws2:Units** indicates the default unit for this property as a string.
270 Other metadata elements, applicable for metric-type properties, are defined in section 3.4.3.

271 2.4.3 Operations

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

275 • *Idempotency* – indicates if invoking the operation twice is equivalent to invoking it once
276 Schema to represent general purpose metadata should be composed from a metadata
277 specification, for example, the WS-Resource Metadata Descriptor [**WSRMD**], as developed in the
278 WS-RF OASIS technical committee.

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

281

```
282 <muws2:PostCondition Dialect="xs:anyURI">  
283   {any} *  
284 </muws2:PostCondition>
```

285 **muws2:PostCondition** contains a statement that asserts "true" immediately after the
286 corresponding operation is complete.

287 **muws2:PostCondition/@muws2:Dialect** is a URI identifying how the statement in
288 *muws2:PostCondition* is built, and what rules govern its evaluation. MUWS defines one possible
289 value for this element. Other values can be defined.

290 The value defined by MUWS is <http://www.w3.org/TR/1999/REC-xpath-19991116>. When this
291 dialect is used, the content of *muws2:PostCondition* is an **[XPath 1.0]** expression. This
292 expression is evaluated against the resource properties document of the manageable resource. If
293 the XPath expression evaluates to a Boolean value of *true*, or, if it evaluates to a non-empty non-
294 boolean value without any errors, then the statement is considered true.

295 2.5 Events

296 2.5.1 Event Format

297 **[MUWS Part 1]** defines the *muws1:ManagementEvent* Global Element Declaration as a container
298 for management events. *muws1:ManagementEvent* allows information to be added via
299 extensibility elements. The *muws2:Situation* element defined below MUST be present as a child
300 of the *muws1:ManagementEvent* element in notifications.

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

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

309 2.5.1.1 Situation Element

310 To support event classifications, the MUWS specification defines the *SituationCategoryType*
311 element, a specialization of a *muws2:CategoryType*. MUWS defines the top level of
312 classifications. Extensions to these classifications enable a refined event classification. Through

313 the use of the extensible *muws2:CategoryType* mechanism, WSDM event consumers can
314 comprehend the situation for an event to a degree commensurate with their ability.

```
315 <muws2:Situation>
316   <muws2:SituationCategory>
317     muws2:SituationCategoryType
318   </muws2:SituationCategory>
319   <muws2:SuccessDisposition>
320     (Successful|Unsuccessful)
321   </muws2:SuccessDisposition> ?
322   <muws2:SituationTime>xs:dateTime</muws2:SituationTime> ?
323   <muws2:Priority>xs:short</muws2:Priority> ?
324   <muws2:Severity>xs:short</muws2:Severity> ?
325   <muws2:Message>muws:LangString</muws2:Message> ?
326   <muws2:SubstitutableMsg MsgId="xs:string" MsgIdType="xs:anyURI">
327     <muws2:Value>xs:anySimpleType</muws2:Value>*
328   </muws2:SubstitutableMsg> ?
329 </muws2:Situation>
```

330 **muws2:Situation/muws2:SituationCategory** categorizes the type of the situation that caused
331 the event report. The values, listed below, represent the names of elements in the *muws2*
332 namespace. The categories are listed in the order of precedence. In a case where there may be
333 some ambiguity about which category to use, the higher precedent category SHOULD be used.
334 The ordering of situation categories is based on empirical data showing relative importance of
335 various types of events. The use of a higher precedent category permits more effective and
336 timely correlation and analysis of events that may indicate the presence of a serious problem.
337 Details and examples for use of the following values are documented in Appendix F. This element
338 is REQUIRED.

- 339 • AvailabilitySituation
- 340 • CapabilitySituation
- 341 • ConfigureSituation
- 342 • StopSituation
 - 343 • **StopInitiated**
 - 344 • **AbortInitiated**
 - 345 • **PauseInitiated**
 - 346 • **StopCompleted**
- 347 • StartSituation
 - 348 • **StartInitiated**
 - 349 • **RestartInitiated**
 - 350 • **StartCompleted**
- 351 • RequestSituation
 - 352 • **RequestInitiated**
 - 353 • **RequestCompleted**
- 354 • DestroySituation
- 355 • CreateSituation
- 356 • DependencySituation
- 357 • ConnectSituation
- 358 • ReportSituation
 - 359 • **Performance**
 - 360 • **Security**
 - 361 • **Heartbeat**
 - 362 • **Status**
 - 363 • **Trace**
 - 364 • **Debug**
 - 365 • **Log**
- 366 • OtherSituation
- 367

368 **muws2:Situation/muws2:SuccessDisposition** in the case where this situation is triggered by a
369 command, this value specifies a successful disposition of the command causing a report of this
370 situation. This element is OPTIONAL and should not be included if the situation is not the result of
371 a command. The element is a restriction of the type `xs:string` allowing the following values:

- 372 • `Successful`
- 373 • `Unsuccessful`

374

375 **muws2:Situation/muws2:SituationTime** represents the date and time an event is observed. If
376 the value does not include a time zone designation, or, if the value does not use 'Z' for UCT, then
377 the value MUST be interpreted as having a time zone of UCT. The value of `SituationTime` MUST
378 provide granularity as precise as supported by the generating platform. This is a REQUIRED
379 element and MUST be provided by the component acting as the originator of an event.

380 **muws2:Situation/muws2:Priority** represents the importance of an event. This element supports
381 management functions requiring an event to be associated with a priority. This is an OPTIONAL
382 element. Values are constrained to a range from 0 through 100. The predefined priorities are:

- 383 • Low (10)
- 384 • Medium (50)
- 385 • High (70).

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

387 **muws2:Situation/muws2:Severity** represents the perceived severity of the status the event is
388 describing with respect to the application that reports the event. This element supports
389 management functions requiring an event to be associated with a severity. This is an OPTIONAL
390 element. Severity levels, based upon the DMTF CIM Alert Indications Perceived Severity, are as
391 follows:

- 392 • 6 (Fatal): a condition is unrecoverable and the service is no longer available.
- 393 • 5 (Critical): a condition affecting the service has occurred. Immediate corrective action is
394 required.
- 395 • 4 (Major): a problem of relatively high severity has occurred. It is likely that normal use of
396 the service is impeded.
- 397 • 3 (Minor): a problem of relatively low severity has occurred. It is unlikely that normal use
398 of the service is impeded.
- 399 • 2 (Warning): a problem affecting the service may occur. Diagnostic and corrective action
400 is recommended.
- 401 • 1 (Information): a message output considered as normal and expected. For example, a
402 process begins, a process finishes, or status information is displayed.
- 403 • 0 (Unknown): a severity level cannot be determined.

404

405 **muws2:Situation/muws2:Message** represents the text accompanying an event. This is typically
406 the resolved message string in a human-readable format, as rendered for a specific locale, and is
407 of type `muws2:LangString` which is an extension of `xs:string` requiring the `xml:lang` attribute. This
408 is an OPTIONAL property. While the string length for `Message` is unbounded, it is
409 RECOMMENDED that the string length for `Message` does not exceed 1024 characters.

410 **muws2:Situation/muws2:SubstitutableMsg** – represents the message data in a substitutable
411 form. The attributes `MsgId` and `MsgIdType` identify the base message type and text. The
412 element value contains the data that will be formatted according to the formatting rules defined by
413 the `MsgId`. This is an OPTIONAL element. However, if this element is used, it must contain all
414 the attributes and elements specified below.

415 **muws2:Situation/muws2:SubstitutableMsg/@muws2:MsgId** specifies the message identifier
416 of an event. This identifier SHOULD be a unique value string, consisting of alphanumeric or
417 numeric characters. The value can be as simple as a string of numeric characters that identify a
418 message in a message catalog. As an alternative, the value can be a multipart string of

419 alphanumeric characters, for example, DBT1234E. This is a REQUIRED attribute. The maximum
420 string length for *MsgId* MUST NOT exceed 256 characters. The *MsgIdType* attribute indicates the
421 formatting type of the *MsgId*.

422 **muws2:Situation/muws2:SubstitutableMsg/@muws2:MsgIdType** specifies the meaning and
423 format of the *MsgId*. This is a REQUIRED attribute. The type of the *MsgIdType* attribute is a URI.

424 **muws2:Situation/muws2:SubstitutableMsg/muws2:Value** can be of any simple type. There
425 are one or more occurrences of this element with each occurrence containing an *xsi:type* attribute
426 defining the type of the contained data. This element is used to pass data values that are
427 substituted as a message is formatted. This element is OPTIONAL. A *MsgId* and *MsgIdType*
428 define rules to map parameters into a composed message, based upon the order of the *Value*
429 elements.

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

```
432 <muws2:Situation>  
433   <muws2:SituationCategory>  
434     <foo:RestartInitiated>  
435       <muws2:StartSituation/>  
436     </foo:RestartInitiated>  
437   </muws2:SituationCategory>  
438   <muws2:SuccessDisposition>Successful</muws2:SuccessDisposition>  
439   <muws2:SituationTime>2004-11-11T18:06:00Z  
440   </muws2:SituationTime>  
441   <muws2:Message xml:lang="en">  
442     Managed Thing XXX: restart processing begun  
443   </muws2:Message>  
444 </muws2:Situation>
```

445 Please note, as outlined in the description of *muws2:CategoryType*, the most general situation
446 classification appears as the innermost element within the XML nest.

447 2.5.1.2 Sequence Number

448 The *sequenceNumber* attribute allows multiple messages to be sent and processed in a logical
449 order that could, owing to idiosyncrasies in the delivery, be different than the order in which they
450 arrived at the consumer. The *sequenceNumber* helps consumers to sort messages into a
451 meaningful flow, as defined by the provider. The *sequenceNumber* is typically used only by
452 event producers when the granularity of the event time stamp (the *reportTime* attribute) is not
453 sufficient to properly sequence events. In other words, the *sequenceNumber* field is typically
454 used to sequence events that have the same time stamp value.

455 It is REQUIRED that *sequenceNumber* for succeeding events be monotonically increasing
456 throughout at least a single value of timestamp.

457 This attribute is OPTIONAL. There is no default value.

```
458 <muws1:ManagementEvent ...  
459   muws2:sequenceNumber="xsd:ulong"? />
```

460

461 2.5.1.3 Event Correlation Properties

462 The optional *EventCorrelationProperties* element allows for the more efficient interpretation and
463 lifecycle management of events. This element is OPTIONAL.

464 **muws1:ManagementEvent/muws1:EventCorrelationProperties:** The
465 *EventCorrelationProperties* element for event correlation follows with the
466 *EventCorrelationPropertiesType* subelements. These properties generally allow for more efficient
467 interpretation and lifecycle management of events. This element is OPTIONAL.

468 **muws1:ManagementEvent/muws1:EventCorrelationProperties/repeatCount:** The
469 `repeatCount` specifies the number of occurrences of identical events within a specified time
470 interval. The time interval is specified by the `elapsedTime` property described next. The
471 definition of “identical events” is application-specific and therefore is not defined by this
472 specification.

473 This property is OPTIONAL and mutable. The `repeatCount` MAY be set by the component
474 that reports the event or by the event consumer. There is no default value. A value of zero or no
475 value indicates no repeated occurrences of the event.

476 **muws1:ManagementEvent/muws1:EventCorrelationProperties/elapsedTime:** The
477 `elapsedTime` is the time interval during which a certain number of identical events occurred.
478 The number of occurrences is specified by the value of `repeatCount`. The `elapsedTime`
479 value indicates the duration of time within which the repeated events were observed.

480 The value of this property MUST be expressed in microsecond granularity.

481 This property is OPTIONAL and mutable; however, if the `repeatCount` is specified, then an
482 elapsed time MUST be present. The `elapsedTime` MUST be set by the same component that
483 sets the `repeatCount`. There is no default value for `elapsedTime`.

484

```
485 <muws1:EventCorrelationProperties ...>  
486   <muws1:repeatCount>xs:short</muws1:repeatCount> ?  
487   <muws1:elapsedTime>xs:long</muws1:elapsedTime> ?  
488 </muws1:EventCorrelationProperties> ?
```

489 2.5.1.4 Message Catalog for Internationalization

490 Late binding of event message text is enabled via `muws2:Situation/muws2:SubstitutableMsg`
491 element described above but the ability to localize message text to a language compatible with
492 the consumer of events may require the ability to reference message catalogs appropriate to the
493 manageable resource reporting the event. The `msgCatalogInformation` element described below
494 provides this additional information.

495 The `msgCatalogInformation` element provides a means to define message catalog information
496 that is to be used for internationalization of messages contained within the event. These
497 properties enable the optional use of a message catalog for specifying messages.

498 This is an OPTIONAL property.

499 The elements within a `msgCatalogInformation` element are:

500 **muws1:ManagementEvent/muws1:msgCatalogInformation/msgCatalog:** The `msgCatalog`
501 element is the qualified URI of the message catalog that contains the locale-dependent message
502 template that is indexed by the `muws-p2-xs:Situation/SubstitutableMsg@MsgId` attribute. The
503 format of the messages in `msgCatalog` is specified by the `msgCatalogType` element.

504 This element is REQUIRED.

505 **muws1:ManagementEvent/muws1:msgCatalogInformation/msgCatalogType:** The
506 `msgCatalogType` URI property specifies the format of the `msgCatalog`. The format defines the
507 substitution identifier syntax for the `muws-p2-xs:Situation/SubstitutableMsg/Value` (that is, the
508 method used to insert runtime information contained in the `muws-p2-`
509 `xs:Situation/SubstitutableMsg/Value` element into the message template that is retrieved from the
510 message catalog to form a completely translated message). The reserved keywords for
511 `msgCatalogType` are:

- 512 • Java- The message catalog uses Java properties encoding. See
513 <http://java.sun.com/j2se/1.4.2/docs/api/java/util/Properties.html>
- 514 • XPG- The message catalog uses X/Open XPG specifications for providing internationalization
515 support. See <http://www.unet.univie.ac.at/aix/aixprgpd/genprogc/nls.htm>

516 Other values may be used for other catalog types.
517 This element is OPTIONAL. Default value is nil and assumed to be understood by the consumer
518 of the event.
519 The schema for *msgCatalogInformation* is:
520

```
521 <muws2:MsgCatalogInformation>  
522   <muws2:msgCatalog>xs:anyURI</muws2:msgCatalog>  
523   <muws2:msgCatalogType>xs:anyURI</muws2:msgCatalogType> ?  
524 </muws2:MsgCatalogInformation>
```

525

526 2.5.2 Topics for capabilities

527 For each capability defined by MUWS, topics are defined that encompass every event related to
528 that capability. For example, if a property related to capability “foo” changes, then a notification is
529 sent to subscribers of the topic corresponding to a change event on this property, as described by
530 [WS-RP]. Concurrently, since this property is associated with the “foo” capability, a notification is
531 also sent to subscribers of the topic encompassing change events associated with capability
532 “foo”.

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

537 The *muws:IdentityCapability* topic defined below is used for events related to the *Identity*
538 capability.

```
539 <wstop:Topic name="IdentityCapability"  
540   messageTypes="muws1:ManagementEvent ">  
541 </wstop:Topic>
```

542 The *muws:ManageabilityCharacteristicsCapability* topic defined below is used for events related
543 to the *ManageabilityCharacteristics* capability.

```
544 <wstop:Topic name="ManageabilityCharacteristicsCapability"  
545   messageTypes="muws1:ManagementEvent ">  
546 </wstop:Topic>
```

547 The *muws:CorrelatablePropertiesCapability* topic defined below is used for events related to the
548 *CorrelatableProperties* capability.

```
549 <wstop:Topic name="CorrelatablePropertiesCapability"  
550   messageTypes="muws1:ManagementEvent ">  
551 </wstop:Topic>
```

552 2.6 Representation of Categorization Taxonomies in XML

553 In the description of several manageability capabilities, categories of information are organized in
554 taxonomies. This is for example the case for the categories of relationships between manageable
555 resources, for operational states of resources, etc. In order to convey category information,
556 including taxonomy lineage, to a manageability consumer, and, in order to represent XML
557 information instances, the following convention is used:

558 MUWS defines an XML Schema complex type called *CategoryType*. The content of XML
559 elements of this type is any XML element. When an element is defined of this type, it MUST obey
560 the following rules:

- 561 • The element and each descendant has, at most, one child element.

- 562
- The top-level element and each descendant represent one category in a taxonomy.
- 563
- The top level element represents the most specialized category. Each element
- 564 represents a more specialized category than the category represented by the element it
- 565 contains, if any.

566 The CategoryType XML Schema type is declared as follows:

```
567 <xs:complexType name="CategoryType">
568 <xs:sequence>
569 <xs:any namespace="##any" minOccurs="0" processContents="lax" />
570 </xs:sequence>
571 </xs:complexType>
```

572 The CategoryType type is used to declare an XML element containing instances of general, or
573 unqualified, category information. The CategoryType type is also used to derive an XML Schema
574 type representing a specific category, for example, a relationship among resources, or among
575 operational states.

576 Category information MUST be declared as follows:

- 577
- An XML element declaring which QName identifies the semantics of the category.
- 578
- The XML element declaring an XML Schema type which is a restriction of
- 579 *muws2:Category*, or a specialized XML Schema type derived from some other refinement
- 580 of *muws2:Category*, for example, *muws2:RelationshipType*.
- 581
- The contents of the XML element MUST be either:
- 582
- The one XML element corresponding to the generalization of the currently declared
- 583 category
- 584
- The empty sequence. This case occurs if the declared category does not have any
- 585 generalizations. For example, the declared category might be the top of a taxonomy.

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

```
589 <mydomain:Off-for-Maintenance>
590 <mydomain:Offline>
591 <anyresource:Unavailable/>
592 </mydomain:Offline>
593 </mydomain:Off-for-Maintenance>
```

594 By processing the XML information, a manageability consumer may learn that a resource is in a
595 state identified by the *mydomain:Off-for-Maintenance* element. However, at the same time, if the
596 manageability consumer is not aware of definitions and semantics associated with the *mydomain*
597 namespace, the consumer may safely assume the resource is in the commonly known state
598 identified by *anyresource:Unavailable*. Since the most specialized elements are first encountered,
599 a consumer can generally stop processing an element of type *muws2:Category* as soon as it
600 reaches an element the semantic of which it understands.

601 3 Capabilities applicable to manageable 602 resources

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

605 3.1 Description

606 The manageability capability URI for the description capability is
607 <http://docs.oasis-open.org/wsdm/muws/capabilities/Description>

608 3.1.1 Definition

609 A manageable resource has a capability description. This description capability gives the
610 manageable resource additional metadata about its capabilities. These attributes include zero or
611 more caption attributes, zero or more description attributes as well as an optional version
612 attribute.

613 3.1.2 Properties

614 This capability defines the following properties:

```
615 <muws2:Caption>muws2:LangString</muws2:Caption> *
```

616 **muws2:Caption** contains a descriptive name for the manageable resource. The *Caption* property
617 is intended for human consumption. A *Caption* is expected to be short and is suitable for display
618 next to a graphic icon. *Caption* is a read-write, optional property with a cardinality of 0 to many.
619 *Caption* is of type *muws2:LangType*, which is a restriction of *xs:string* carrying an *xml:lang*
620 attribute. This attribute contains a language identifier as defined by [RFC3066]. There can not be
621 more than one *Caption* per language identifier.

622 Metadata for *Caption*:

623 It is *Mutable*

624 It is *Modifiable*

625 It has the following *Capability* metadata item:

```
626 <muws2:Capability>  
627   http://docs.oasis-open.org/wsdm/muws/capabilities/Description  
628 </muws2:Capability>
```

629

```
630 <muws2:Description>muws2:LangString</muws2:Description> *
```

631 **muws2:Description** is a string containing a description for the resource being managed. The
632 *Description* property is intended for human consumption. A *Description* is expected to be longer
633 and more detailed than a *Caption*. *Description* is a read-write optional property with a cardinality
634 of 0 to many. *Description* is of type *muws2:LangType*, which is a restriction of *xs:string* carrying
635 an *xml:lang* attribute. This attribute contains a language identifier as defined by [RFC3066].
636 There cannot be more than one *Description* per language identifier.

637 Metadata for *Description*:

638 It is *Mutable*

639 It is *Modifiable*

640 It has the following *Capability* metadata item:

```
641 <muws2:Capability>
642   http://docs.oasis-open.org/wsdm/muws/capabilities/Description
643 </muws2:Capability>
```

644

```
645 <muws2:Version>xs:string</muws2:Version> ?
```

646 **muws2:Version** is a string representing the version of the resource being managed. MUWS
647 does not specify how this string is constructed. The *Version* string can be specified by any
648 domain-specific specification that uses MUWS. *Version* is an optional property with a cardinality
649 of 0 to 1.

650 Metadata for *Version*:

651 It is *Mutable*

652 It is *Modifiable*

653 It has the following *Capability* metadata item:

```
654 <muws2:Capability>
655   http://docs.oasis-open.org/wsdm/muws/capabilities/Description
656 </muws2:Capability>
```

657 3.1.3 Events

658 The *muws:DescriptionCapability* topic defined below is used for events related to the *Description*
659 capability.

```
660 <wstop:Topic name="DescriptionCapability"
661             messageTypes="muws1:ManagementEvent">
662 </wstop:Topic>
```

663 3.2 State

664 The manageability capability URI for the State capability is
665 <http://docs.oasis-open.org/wsdm/muws/capabilities/State>

666 3.2.1 Definition

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

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

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

679 3.2.2 Describing State Models

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

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

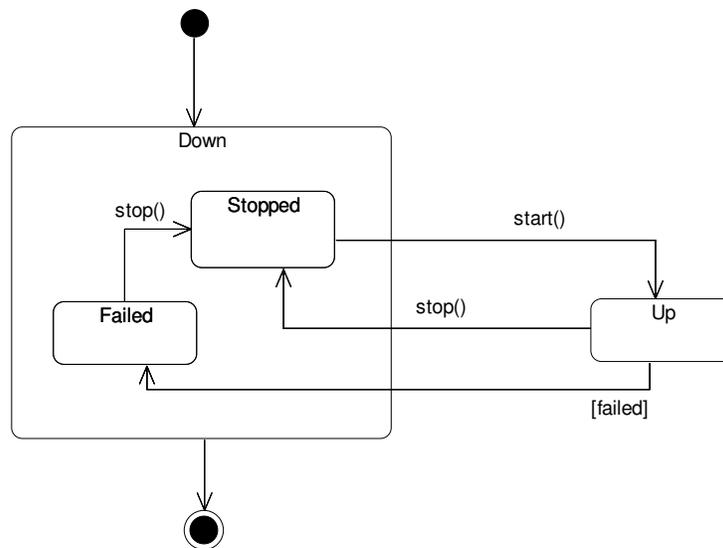
685 States can have sub-states that MUST be wholly contained within a higher-level state.

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

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

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

695 Figure 1 shows a simple state model that is used as an example in this section – it does not
 696 constitute the specification of a recommended state model.



697
 698

Figure 1: Example Operational State Model

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

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

- 703 • *exns:Down*
 704 This QName corresponds to the “Down” state in the UML diagram. A resource in this
 705 state is unable to perform any of its functional tasks.
- 706 • *exns:Stopped*
 707 This QName corresponds to the “Stopped” sub-state of the “Down” state in the UML
 708 diagram. Since this state is a sub-state of the “Down” state, it follows that a resource in
 709 the “Stopped” sub-state is unable to perform any of its functional tasks. A manageable
 710 resource exposing this state model can be started from the “Stopped” sub-state.
- 711 • *exns:Failed*
 712 This QName corresponds to the “Failed” sub-state of the “Down” state in the UML diagram.
 713 Since this state is a sub-state of the “Down” state, it follows that a resource in the “Failed”

714 sub-state is unable to perform any of its functional tasks. A manageable resource
715 exposing this state model can not be started directly from the “Failed” sub-state. Such a
716 resource must first transition to the “Stopped” sub-state.

- 717 • *exns:Up*
718 This QName corresponds to the “Up” state in the UML diagram. A resource in this state is
719 able to perform at least some of its functional tasks.

720 3.2.3 Information Markup Declarations

721 3.2.3.1 Representation of States

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

728 *muws2:StateType* XML Schema type is declared as follows

```
729 <xs:complexType name="StateType">  
730 <xs:complexContent>  
731 <xs:extension base="muws2:CategoryType" />  
732 </xs:complexContent>  
733 </xs:complexType>
```

734 The *muws2:StateType* type is used to declare an XML element containing an instance of state.

735 A state MUST be declared as follows:

- 736 • An XML element declaring which QName identifies the semantics of the state.
- 737 • The XML element has an XML Schema type of *muws2:StateType*, or a restriction of
738 *muws2:StateType*.
- 739 • The contents of the XML element MUST be either:
 - 740 • The one XML element that corresponds to the state containing this state. In other
741 words, this state is a sub-state of another state.
 - 742 • The empty sequence. This case occurs if this state is not a sub-state of another
743 state.

744 For example, the “Failed” state in the example above is a sub-state of the “Down” state. An
745 instance of the “Failed” state may be represented, using the rules described above, by the
746 following XML fragment:

```
747 <my:StateTypeInstanceElement xsi:type="StateType">  
748 <exns:Failed>  
749 <exns:Down/>  
750 </exns:Failed>  
751 </my:StateTypeInstanceElement>
```

752 3.2.3.2 Representation of state

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

```
755 <muws2:State>muws2:StateType</muws2:State>
```

756 The State element provides a representation of the state of a manageable resource. The State
757 element follows the convention for the *muws2:CategoryType* type described in section 2.6. This
758 convention allows the rendering of a hierarchy of states and sub-states. State values are defined

759 in the operational state model for the resource. This specification does not define the operational
760 state model for any resource.

761 3.2.3.3 Representation of state transition

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

```
764 <muws2:StateTransition Time"xs:dateTime"  
765             TransitionIdentifier=" xs:anyURI"?>  
766   <muws2:EnteredState>muws2:StateType</muws2:EnteredState>  
767   <muws2:PreviousState>muws2:StateType</muws2:PreviousState?>  
768   {any} *  
769 </muws2:StateTransition>
```

770 **muws2:StateTransition** is used for representing information about a state change.

771 **muws2:StateTransition/@muws2:Time** attribute indicates the time at which the transition
772 occurred (transitions are assumed to be instantaneous). This attribute is REQUIRED.

773 **muws2:StateTransition/@muws2:TransitionIdentifier** attribute indicates the actual transition
774 that occurred. This attribute is OPTIONAL and may be omitted where, for example, there is only
775 one transition between the *EnteredState* and the *PreviousState*.

776 **muws2:StateTransition/muws2:EnteredState** element indicates which state has been entered
777 during the transition. This element is REQUIRED.

778 **muws2:StateTransition/muws2:PreviousState** element indicates the state that the resource
779 was in immediately prior to the state change occurring. This element is OPTIONAL to allow for
780 the time between the state model being created in some initial state, for example when the
781 resource is created, and the time of the transition from that initial state.

782 3.2.4 Properties

783 This capability does not define any standard property.

784 A capability defining a state model SHOULD define a resource property that exposes the state. It
785 is RECOMMENDED that a state model also define a resource property that exposes the last
786 state transition.

787 The property used to expose the state must either contain the *muws2:State* element or be of type
788 *muws2:StateType*. The name of the property can be any name meaningful to the state model
789 defined in the capability. There may be multiple state capabilities, and therefore multiple state
790 properties for a resource. The metadata for this property SHOULD include the possible values.
791 That is, the state model should provide a list of states in the state model.

792 The property to represent the last transition, if such a property is provided, must contain the
793 element *muws2:StateTransition*. The name of the last transition property can be any name
794 meaningful to the state model. There may be multiple state capabilities and multiple properties
795 exposing the last transition.

796 3.2.4.1 Example

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

```
798 <foo:OperationalState>  
799   <muws2:State>...</muws2:State>  
800 </foo:OperationalState>  
801 <foo>LastOperationalStateTransition>  
802   <muws2:StateTransition>...</muws2:StateTransition>  
803 </foo>LastOperationalStateTransition?>
```

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

```

806 <foo:OperationalState>
807   <muws2:State>
808     <exns:Failed><exns:Down/></exns:Failed>
809   </muws2:State>
810 </foo:OperationalState>
811 <foo>LastOperationalStateTransition>
812   <muws2:StateTransition Time="2004-03-11T11:30:56Z"
813 TransitionIdentifier="http://example.com/SimpleOperationalState/T/Failed">
814   <muws2:EnteredState>
815     <exns:Failed><exns:Down/></exns:Failed>
816   </muws2:EnteredState>
817   <muws2:PreviousState>
818     <exns:Up/>
819   </muws2:PreviousState>
820 </muws2:StateTransition>
821 </foo>LastOperationalStateTransition>

```

822 In this example, the *foo:OperationalState* property contains the current operational state of the
823 resource, using the *muws2:State* element defined in section 3.2.3.2 The
824 *foo>LastOperationalStateTransition* property contains a description of the most recent operational
825 state transition for the resource, using the *muws2:StateTransition* element as defined in section
826 3.2.3.2.

827 3.2.5 Operations

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

831 3.2.6 Events

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

```

833 <wstop:Topic name="StateCapability"
834           messageTypes="muws1:ManagementEvent ">
835 </wstop:Topic>

```

836 It is RECOMMENDED that resources send a notification on a transition between states. The topic
837 defined for the *State* capability SHALL be used to publish such notifications. If a resource sends
838 such a notification, then the notification message MUST contain at least the XML element
839 representing a state transition (*muws2:StateTransition*).

840 To obtain events about a certain state transition, a subscriber can use a *Selector*, on the
841 notification subscription, to select only those events containing the required
842 *muws2:TransitionIdentifier* element in the notification content, or, a combination of
843 *muws2:EnteredState* and *muws2:PreviousState* elements in the notification content. The *Selector*
844 mechanism is described in [WSN].

845 To filter for events about entry into a particular state or set of states, a *Selector* expression based
846 on the *muws2:EnteredState* element can be used. To filter for events about exit from a particular
847 state or set of states a *Selector* expression based on the *muws2:PreviousState* element can be
848 used.

849 3.3 Operational Status

850 The manageability capability URI for this capability is
851 <http://docs.oasis-open.org/wsdm/muws/capabilities/OperationalStatus>

852 3.3.1 Definition

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

861 3.3.2 Properties

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

```
863 <muws2:OperationalStatus>  
864   (Available|PartiallyAvailable|Unavailable|Unknown)  
865 </muws2:OperationalStatus>
```

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

```
868 <muws2:OperationalStatus>Available</muws2:OperationalStatus>
```

869 The *muws2:OperationalStatus* property is of type *muws2:OperationalStatusType*. The type is a
870 restriction of *xs:string* and provides a simple indication of the availability of the resource,
871 independent of the potentially complex operational state model. This property has a cardinality of
872 1. The valid values are:

- 873 • *Available*: This value indicates that a manageable resource is operating normally within
874 any configured operating parameters, and is able to perform all functional tasks.
- 875 • *PartiallyAvailable*: This value indicates that a manageable resource is operating, but
876 outside of configured operating parameters. A manageable resource reporting this
877 operational status is able to perform some, but not all, functional tasks. A manageable
878 resource may, for example, be in the process of starting or a resource may be lacking
879 some resource it needs to perform.
- 880 • *Unavailable*: This value indicates that a manageable resource is not operating, and is not
881 able to perform any functional tasks. A manageable resource may have been stopped,
882 or may have failed.
- 883 • *Unknown*: This value indicates that a manageable resource is unable to report status at
884 this time.

885 Metadata for *OperationalStatus*:

886 It is *Mutable*

887 It is not *Modifiable*

888 It has the following *Capability* metadata item:

```
889 <muws2:Capability>  
890   http://docs.oasis-open.org/wsdm/muws/capabilities/OperationalStatus  
891 </muws2:Capability>
```

892 3.3.3 Events

893 The *muws:OperationalStatusCapability* topic defined below is used for events related to the
894 *Operational Status* capability.

```
895 <wstop:Topic name="OperationalStatusCapability"  
896   messageTypes="muws1:ManagementEvent">  
897 </wstop:Topic>
```

898 No specific event is defined, since the notification on property value change provided by WS-
899 ResourceProperties is sufficient, when applied to the *muws2:OperationalStatus* property.

900 3.4 Metrics

901 The manageability capability URI for this capability is
902 <http://docs.oasis-open.org/wsdm/muws/capabilities/Metrics>

903 3.4.1 Definition

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

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

916 The following sections define two (2) types of “metadata” used for metrics. The *value modifiers*
917 provides context to the metric’s value, improving the consumer’s ability to interpret the value.

918 The *definitional metadata* describes the means by which the value of the metric is acquired.

919 It is important to keep the two distinct types of metadata or data modifiers in mind when reading
920 the following sections as it is sometimes the case that there are two similar definitions,
921 distinguished primarily by the entity (value or metric) that they describe.

922 Note: Future revisions of this specification are likely to move this definitional metadata into an
923 external metadata component, since it is used to define the configuration of the metric properties.
924 However, the value modifiers will remain associated with the metric itself, as their values is
925 directly pertinent to the interpretation of the metric value.

926 3.4.2 Information Markup Declarations

927 The following schema fragment declares the (reusable) data type used to expose the metrics of a
928 resource. All attributes defined in the *muws2:MetricAttributes* attribute group are OPTIONAL.
929 The *MetricAttributes* group defines the value modifiers associated with the metric.

```
930 <xs:attributeGroup name="MetricAttributes">  
931   <xs:attribute name="ResetAt" type="xs:dateTime"/>  
932   <xs:attribute name="LastUpdated" type="xs:dateTime"/>  
933   <xs:attribute name="Duration" type="xs:duration"/>  
934 </xs:attributeGroup>
```

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

937 **(MetricAttributes)/ResetAt** indicates the time when a metric value was reset. See the definition
938 of *muws2:TimeScope* for information on when to provide this attribute. If the attribute value does
939 not include a time zone indication, or Z for UTC, then the value MUST be interpreted as UTC.

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

943 **(MetricAttributes)/Duration** indicates the measurement period relative to the value of the
944 *LastUpdated* metric attribute. This is the time over which a metric value was collected, counted,
945 or measured.

946
947 The *Duration* attribute is valid only for a metric that covers a time interval. Therefore, the
948 *Duration* attribute **MUST** be included for a metric having a *TimeScope* of *Interval*. Similarly, it
949 **MUST NOT** be included for a metric having a *TimeScope* of *PointInTime* or *SinceReset*, because
950 these *TimeScopes* imply their measurement period. For these *TimeScopes*, an implementer
951 should make use of *ResetTime* and *CurrentTime* to calculate the duration for the collection of a
952 metric value.

953 The following metric type definition is an example of how a metric attribute is incorporated into a
954 metric type. All metric types **MUST** incorporate the *muws2:MetricAttributes* attribute group.

```
955 <xs:complexType name="MyExampleIntegerMetricType">  
956   <xs:simpleContent>  
957     <xs:extension base="xs:integer">  
958       <xs:attributeGroup ref="muws2:MetricAttributes"/>  
959       <xs:anyAttribute namespace="##other" processContents="lax"/>  
960     </xs:extension>  
961   </xs:simpleContent>  
962 </xs:complexType>
```

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

```
964 <MyIntegerMetric  
965   LastUpdated="2004-03-11T11:30:56Z"  
966   Duration="PT1H">  
967   12345  
968 </MyIntegerMetric>
```

969 3.4.3 Metadata

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

971 It is *Mutable*

972 It is not *Modifiable*

973 It has the following *Capability* metadata item:

```
974 <muws2:Capability>  
975   http://docs.oasis-open.org/wsdm/muws/capabilities/Metrics  
976 </muws2:Capability>
```

977 In addition to the value modifiers, the following additional metadata items are defined for a
978 property that is a metric. As opposed to the value modifiers described above, this metadata
979 describes the metric itself, independent of the value at any particular time.

```
980 <muws2:ChangeType> (Counter | Gauge | Unknown) </muws2:ChangeType>
```

981 **muws2:ChangeType** is an enumeration indicating how a change to an associated metric value
982 should be interpreted by a consumer. A property representing a metric **MUST** include a single
983 instance of *ChangeType* in its metadata description. Each *ChangeType* value is interpreted as
984 follows:

- 985 • *Counter* - the value of the metric is a monotonically increasing integer. Such a metric
986 value increases by increments of "1" when a situational event occurs to the resource.
- 987 • *Gauge* – changes of the value of the metric are not constrained in the way changes to
988 *Counter* metrics are constrained.
- 989 • *Unknown* - the change behavior for the value of the metric is not known or cannot be
990 described.

991

```
992 <muws2:TimeScope>
993   (Interval|PointInTime|SinceReset)
994 </muws2:TimeScope>
```

995 **muws2:TimeScope** is an enumeration for indicating if there is some interval, over which the data
996 is collected, counted, or measured. A property that is a metric MUST include a single instance of
997 *TimeScope* in its metadata description. Each *TimeScope* value is interpreted as follows:

- 998 • *Interval* - the value of a metric is collected over some time interval. In this case a *Duration*
999 attribute MUST be reported with a metric property. The value of a *Duration* attribute is the
1000 elapsed time, from the beginning of an interval, to the end of an interval. A *Duration* MAY
1001 change, but usually remains the same for every collection of a metric. The *ResetAt*
1002 attribute MAY also be reported with such a metric property and may have a value during
1003 the measurement period.
- 1004 • *PointInTime* - the value of a metric is counted, collected, or measured at a single instant
1005 in time. In this case a *Duration* attribute MUST NOT be reported with a metric property.
1006 A metric defined with a *TimeScope* of *PointInTime* does not support a reset capability
1007 and MUST NOT include a *ResetAt* attribute.
- 1008 • *SinceReset* - the value of the metric is collected since the last reset of a resource, or
1009 since the manageable resource started collecting data for a metric. . In this case a
1010 *Duration* attribute MUST NOT be reported with a metric property, and a *ResetAt* attribute
1011 MUST be reported.

1012

```
1013 <muws2:GatheringTime>
1014   (OnChange|Periodic|OnDemand|Unknown)
1015 </muws2:GatheringTime>
```

1016 **muws2:GatheringTime** is an enumeration indicating under which circumstance the value of a
1017 metric is updated. A property that is a metric MUST include a single instance of
1018 *muws2:GatheringTime* in its metadata description. Each *muws2:GatheringTime* value is
1019 interpreted as follows:

- 1020 • *OnChange* - the value of a metric is updated whenever a change occurs to the quantity
1021 measured.
- 1022 • *Periodic* - the value of a metric is updated on a regularly scheduled basis.
- 1023 • *OnDemand* - the value of a metric is updated when processing a request for the metric
1024 value.
- 1025 • *Unknown* - it is unknown when the value of a metric is updated.

1026

```
1027 <muws2:CalculationInterval>xs:duration</muws2:CalculationInterval>
```

1028 **muws2:CalculationInterval** represents the interval at which a value of a metric is gathered or
1029 calculated by a resource. The value of a metric is not updated during a calculation interval. Unlike
1030 *Duration*, which can change every time the metric is updated, the value of *CalculationInterval* is
1031 expected to change rarely. This is because *CalculationInterval* is used only for a value of a
1032 metric that is updated at regular intervals.

1033 Note also that it is possible for a *CalculationInterval* to be different than the *Duration*. The former
1034 specifies the frequency of update while the latter specifies the period over which the data has
1035 been collected.

1036

```
1037 <muws2:MetricGroup>xs:anyURI</muws2:MetricGroup>
```

1038 **muws2:MetricGroup** indicates that a metric property is a member of a group of metrics. A metric
1039 property MAY be a member of zero or more metric groups. A metric group is identified by a URI.
1040 Each metric property included in a metric group MUST have a *muws2:MetricGroup* element

1041 containing an identical URI. A metric property MAY include zero or more *muws2:MetricGroup*
 1042 elements in its metadata description. Each *muws2:MetricGroup* element represents a
 1043 membership of the metric property in a metric group.

1044 3.4.3.1 An Example

1045 To illustrate how the value modifiers and definitional metadata might work together, consider a
 1046 metric that computes average request size over the previous hour. Such a metric might have the
 1047 following description. (Note that this example uses human readable rather than XML data
 1048 representations.)

1049

Definitional Metadata	
ChangeType	Gauge
TimeScope	Interval
GatheringTime	Periodic
CalculationInterval	1 Minute
Value Modifiers	
ResetAt	2:30pm
Duration	1 Hour
LastUpdated	4:00pm
(value)	4800

1050

1051 These values describe our average request size that is a gauge that is updated once a minute to
 1052 produce a sliding scale one hour in length. This instance shows an average size of 4800 from 3-
 1053 4pm.

1054 3.4.4 Properties

1055 The following fragment provides the specification of a resource metrics property:

1056 `<muws2:CurrentTime>xs:dateTime</muws2:CurrentTime>`

1057 **muws2:CurrentTime** contains the current time, as known to a resource, when a property was
 1058 retrieved from a manageable resource. This property is useful to a manageability consumer in the
 1059 absence of a time synchronization mechanism when analyzing the time values received from a
 1060 manageability endpoint. *muws2:CurrentTime* is a read-only mandatory property with a resource
 1061 cardinality of 1.

1062 The Metrics capability requires the *muws2:CurrentTime* property to be present in a resource
 1063 property. The *muws2:CurrentTime* property provides a reference point for time-based attributes,
 1064 as defined by metric data types. Note that *muws2:CurrentTime* is not a metric. Rather, it is a
 1065 property of type *xs:dateTime* defined as part of the "Metrics" capability, consequently, any reset
 1066 operations has no effect on *muws2:CurrentTime*.

1067 3.4.5 Events

1068 The *muwse:MetricsCapability* topic defined below is used for events related to the *Metrics*
 1069 capability.

```
1070 <wstop:Topic name="MetricsCapability"  
1071         messageTypes="muws1:ManagementEvent ">  
1072 </wstop:Topic>
```

1073 WS-ResourceProperties specifies the ability to define optional topics for a resource property that
1074 can emit notifications when a value changes. These topics allow a consumer to request
1075 notifications on an update of a metric property.

1076 3.5 Configuration

1077 The manageability capability URI for this capability is
1078 <http://docs.oasis-open.org/wsdm/muws/capabilities/Configuration>

1079 3.5.1 Definition

1080 A configuration property is any resource property exposing a value that, when changed, changes
1081 some operational behavior of the resource.

1082 The value of a configuration property may be changed directly by a set operation, or, may be
1083 changed as a side effect of some other operation.

1084 3.5.2 Properties

1085 MUWS does not define any required property for the *Configuration* capability. Domain experts
1086 can define configuration properties which are then marked as associated with the configuration
1087 capability. The metadata for a configuration property MUST be:

1088 It is *Mutable*

1089 It is *Modifiable* only if the WS-ResourceProperties *SetResourceProperty* operation can be used to
1090 change the value of the property. It is not *Modifiable* if the property is changed only as a side
1091 effect.

1092 It has the following *Capability* metadata item:

```
1093 <muws2:Capability>  
1094   http://docs.oasis-open.org/wsdm/muws/capabilities/Configuration  
1095 </muws2:Capability>
```

1096 3.5.3 Operations

1097 WS- ResourceProperties *SetResourceProperty* operation MAY be used to change a configuration
1098 value.

1099 3.5.4 Events

1100 The *muws:ConfigurationCapability* topic defined below is used for events related to the
1101 *Configuration* capability.

```
1102 <wstop:Topic name="ConfigurationCapability"  
1103         messageTypes="muws1:ManagementEvent ">  
1104 </wstop:Topic>
```

1105 4 Capabilities applicable to management in 1106 general

1107 Section 3, "Capabilities applicable to manageable resources", when merged with the capabilities
1108 defined in **[MUWS Part 1]**, provide the list of manageability capabilities defined by MUWS. This
1109 section provides management-related capabilities that are different from manageability
1110 capabilities.

1111 A *manageability capability* is offered by a manageability representation and a manageability
1112 capability applies to a resource as represented by a manageability representation. In contrast, a
1113 management-*related capability* can be offered by any endpoint of a Web service, not just a
1114 manageability endpoint.

1115 The function of a management-related capability is related to the management of a resource, but
1116 it is not necessarily offered directly by a manageability endpoint of a resource. For example, the
1117 capability to help a manageability consumer discover a new manageable resource can be
1118 provided by a registry instead of by a management representation of the resource. As another
1119 example, a manageable resource may provide information about relationships in which it
1120 participates. The information about a relationship may also provide valid information for another
1121 entity or resource that is not manageable, like a registry, maintaining and providing relationship
1122 information about a resource without the resource providing the relationship information directly.

1123 4.1 Relationships

1124 The manageability capability URI for this capability is
1125 <http://docs.oasis-open.org/wsdm/muws/capabilities/Relationships>

1126 4.1.1 Definition

1127 A relationship is an N-ary association between resources. A relationship may have properties,
1128 operations and other characteristics. One of these properties is a type that conveys the semantic
1129 of the relationship. The resources involved in the relationship are called participants. Each
1130 participant has a role in the relationship. The participants may or may not be manageable
1131 resources in the MUWS sense. The notion of "direction" of a relationship is a semantic
1132 interpretation based on role definitions. There could be many instances of relationships between
1133 many instances of resources.

1134 Note that this capability is not limited to manageable resources and can be exposed by any
1135 resource that wants to expose relationships that it knows about.

1136 A relationship may become stale. The information about a relationship should be validated, either
1137 manually or automatically, before it can be relied upon. Exposing the information about a
1138 relationship should be considered a potential security risk if a participating resource should not be
1139 visible for security reasons.

1140 4.1.2 Information Markup Declarations

1141 4.1.2.1 Representation of Categories of Relationships

1142 A relationship may be categorized as a certain type of relationship. A relationship type defines the
1143 semantics of the relationship. One relationship type may be a specialization or generalization of
1144 another type. This defines a taxonomy of relationship categories. MUWS defines a way to
1145 represent a type and its taxonomy lineage, but the actual definition of a relationship type is
1146 specific to a resource management model. Therefore, no relationship type is defined by MUWS.

1147 In other words, MUWS specifies only the mechanism to convey a relationship type, or category,
1148 in XML as follows.

1149 *RelationshipTypeType* type is declared as follows

```
1150 <xs:complexType name="RelationshipTypeType">  
1151   <xs:complexContent>  
1152     <xs:extension base="muws2:CategoryType"/>  
1153   </xs:complexContent>  
1154 </xs:complexType>
```

1155 The *RelationshipTypeType* type is used to declare an XML element containing instances of
1156 relationship type information.

1157 The relationship type information MUST be declared as follows:

- 1158 • An XML element declaring which QName identifies the semantics of a relationship type.
- 1159 • The XML element MUST be declared with an XML Schema type that is a restriction of
1160 *RelationshipTypeType*.
- 1161 • The contents of the XML element MUST be either
 - 1162 • The only one XML element corresponding to the generalization of the currently
1163 declared relationship type.
 - 1164 • The empty sequence, if the currently declared relationship type does not have a
1165 generalization, such as the top of a taxonomy.

1166 For example, the “USB attached” relationship type may be generalized to the “Bus connected”
1167 type which, in turn, may be generalized to the “Generally linked” type. An instance of the “USB
1168 attached” relationship type information may be represented in the following XML fragment by
1169 using the rules described above:

```
1170 <my:RelationshipTypeInstanceElement xsi:type="RelationshipTypeType">  
1171   <usb:Attached>  
1172     <bus:Connected>  
1173       <generally:Linked/>  
1174     </bus:Connected>  
1175   </usb:Attached>  
1176 </my:RelationshipTypeInstanceElement>
```

1177 4.1.2.2 Representation of an Instance of a Relationship

1178 MUWS defines the following Global Element Declaration (GED) to represent an instance of a
1179 relationship.

```
1180 <muws2:Relationship>  
1181   <muws2:Name>xs:string</muws2:Name> ?  
1182   <muws2:Type>muws2:RelationshipTypeType</muws2:Type>  
1183   <muws2:Participant>  
1184     <muws2:Self/> ?  
1185     <muws1:ManageabilityEndpointReference/> *  
1186     <wsa:EndpointReference/> *  
1187     <muws1:ResourceId/> ?  
1188     <muws2:Role>xs:anyURI</muws2:Role>  
1189     {any} *  
1190   </muws2:Participant>  
1191   <muws2:Participant/>+  
1192   <muws2:AccessEndpointReference>  
1193     wsa:EndpointReferenceType  
1194   </muws2:AccessEndpointReference>?  
1195   {any} *  
1196 </muws2:Relationship>
```

1197 **muws2:Relationship/muws2:Name** is a human readable name for a relationship. *Name* should
1198 not be used for machine reasoning about the semantics of a relationship. Type should be used
1199 instead. This element is OPTIONAL.

1200 **muws2:Relationship/muws2:Type** is the relationship type this relationship belongs to.
1201 Examples of such types include linkage, containment, or dependency. MUWS does not define
1202 any specific relationship type. This is left to domain-specific models. MUWS only defines a way to
1203 convey the type as part of the representation of a relationship. In order to allow relationships to be
1204 defined as part of a taxonomy, the mechanism used by MUWS to represent relationship types
1205 leverages the *muws2:CategoryType* type defined in section 2.6. This element is REQUIRED.

1206 **muws2:Relationship/muws2:Participant** contains information about a participant in the
1207 relationship. There **MUST** be at least two participants, but there **MAY** be more than two
1208 participants.

1209 **muws2:Relationship/muws2:Participant/muws2:Self**: is an optional empty element that allows
1210 a participant to reference its self as the participant.

1211 **muws2:Relationship/muws2:Participant/muws1:ManageabilityEndpointReference** is a
1212 reference to a WSDM manageability endpoint. This GED is defined in part 1. It **MAY** be included
1213 if a participant is a WSDM manageable resource and the provider wishes to expose this
1214 information. If more than one manageability endpoint is known, then more than one instance of
1215 this element **MAY** be present.

1216 **muws2:Relationship/muws2:Participant/wsa:EndpointReference** is a WS-Addressing
1217 endpoint Reference.

1218 **muws2:Relationship/muws2:Participant/muws1:ResourceId** is a WSDM manageable
1219 resource identifier which **MAY** be reported by the provider of relationship information. This GED is
1220 defined in part 1. This information may be used to locate manageability endpoints for a
1221 participant, or may be used for other purposes. For example, a resource identifier **SHOULD** be
1222 used to express that the provider of relationship information is also a participant in a relationship
1223 by returning its own resource identifier as one of the participants. Obviously, in order for this
1224 assertion to work, the provider of relationship information must be a WSDM manageable
1225 resource.

1226 **muws2:Relationship/muws2:Participant/muws2:Role** is a URI which identifies the role a
1227 participant plays in a relationship. A participant role **MUST** be unique within a given instance of
1228 the relationship. The set of valid roles is defined by a relationship type. This attribute is
1229 REQUIRED.

1230 **muws2:Relationship/muws2:Participant/{any}*** is an XML extensibility content which **MAY**
1231 contain elements that further or otherwise describe a participant.

1232 **muws2:Relationship/muws2:AccessEndpoint** is a reference to a Web service endpoint which
1233 provides access to this relationship (if available). The endpoint **MUST** implement the relationship
1234 resource capability (see section 4.2).

1235 The following is an example of a relationship information instance. The relationship is a WSDM
1236 manageable network host myhost.myorg.org containing an attached SCSI disk. The SCSI disk is
1237 exposed as a functional or operational endpoint of a Web service (e.g. to read/write from the
1238 disk). The "containment" relationship is represented by the following XML instance fragment:

```

1239 <muws2:Relationship>
1240   <muws2:Name>SCSI disk attached to the host computer</muws2:Name>
1241   <muws2:Type>
1242     <scsi:Attached>
1243       <bus:Connected>
1244         <generally:Linked/>
1245       </bus:Connected>
1246     </scsi:Attached>
1247   </muws2:Type>
1248   <muws2:Participant>
1249     <muws1:ManageabilityEndpointReference>

```

```

1250     ...EPR1...
1251     </muws1:ManageabilityEndpointReference>
1252     <wsa:EndpointRefence>
1253     ...EPR2...
1254     </wsa:EndpointReference>
1255     <muws1:ResourceId>urn:uuid:123</muws1:ResourceId>
1256     <muws2:Role>urn:role:bus:host</muws2:Role>
1257     <netop-xs:HostName>myhost.myorg.org</netop-xs:NostName>
1258 </muws2:Participant>
1259 <muws2:Participant>
1260   <muws2:Self/>
1261   <muws2:Role>urn:role:bus:device</muws2:Role>
1262   <scsi-xs:Port>2</scsi-xs:Port>
1263   <scsi-xs:CH>0</scsi-xs:CH>
1264   <scsi-xs:BusID>5</scsi-xs:BusID>
1265   <scsi-xs:LUN>0</scsi-xs:LUN>
1266 </muws2:Participant>
1267 </muws2:Relationship>

```

1268 4.1.3 Properties

1269 The Relationship capability defines the following property:

```
1270 <muws2:Relationship/> *
```

1271 **muws2:Relationship** is a representation of a relationship of which the provider of this capability
1272 is aware. See section 4.1.2.2 for the definition of the Relationship element. The provider of this
1273 capability is not necessarily a participant in any relationship represented by this property.

1274 It is not recommended to request all values of the Relationship property with either *wsrf-*
1275 *rp:GetResourceProperty* or *wsrf-rp:GetMultipleResourceProperties* operations as there may be
1276 too many relationships. The use of the *wsrf-rp:QueryResourceProperties* operation is
1277 RECOMMENDED when retrieving the Relationships property. A provider of this manageability
1278 capability SHOULD, in general, support the *wsrf-rp:QueryResourceProperties* operation.
1279 However, if the provider of this capability knows of just a few relationships, it MAY choose not to
1280 support *wsrf-rp:QueryResourceProperties* operation.

1281 For example, the following request may be sent to retrieve all “Bus connected” relationships
1282 which point to devices exposed as Web services.

```

1283 <soap:Envelope ...>
1284   <soap:Header>
1285     ...
1286   </soap:Header>
1287   <soap:Body>
1288     <wsrf-rp:QueryResourceProperties>
1289       <wsrf-rp:QueryExpression
1290         Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116" >
1291         boolean(/*/muws2:Relationship/muws2:Type/*/bus:Connected and
1292         /*/muws2:Relationship/muws2:Participant [Role="urn:role:bus:device"]/mows-
1293         xs:EndpointReference)
1294       </wsrf-rp:QueryExpression>
1295     </wsrf-rp:QueryResourceProperties>
1296   </soap:Body>
1297 </soap:Envelope>

```

1298 4.1.4 Operations

1299 This capability defines the following message exchanges.

1300 4.1.4.1 QueryRelationshipsByType

1301 This operation is OPTIONAL. It is a shortcut to query relationships of the same type. The request
1302 to perform this operation has a payload as follows:

```
1303 <muws2:QueryRelationshipsByType>  
1304   <muws2:RequestedType>xs:QName</muws2:RequestedType> +  
1305 </muws2:QueryRelationshipsByType>
```

1306 **muws2:QueryRelationshipsByType** is a Global Element Declaration (GED) which identifies the
1307 operation requested.

1308 **muws2:QueryRelationshipsByType/muws2:RequestedType** is a QName which identifies the
1309 requested type(s) of relationship(s). When processing this request, the manageability endpoint
1310 MUST return any available instance relationship that is of the requested type or of any type that is
1311 a specialization of the requested type. There can be more than one requested type, in which case
1312 any relationship instance corresponding to any requested type MUST be returned.

1313 The response to the above request is either a fault (any fault) or the following message:

```
1314 <muws2:QueryRelationshipsByTypeResponse>  
1315   <muws2:Relationship/> *  
1316 </muws2:QueryRelationshipsByTypeResponse>
```

1317 **muws2:QueryRelationshipsByTypeResponse** is a GED which identifies a response to the
1318 requested operation.

1319 **muws2:QueryRelationshipsByTypeResponse/muws2:Relationship** is a relationship
1320 representation matching a requested type. There is one such element for each relationship
1321 instance corresponding to at least one requested type.

1322 This operation has the following *Capability* metadata item:

```
1323 <muws2:Capability>  
1324   http://docs.oasis-open.org/wsdm/muws/capabilities/Relationships  
1325 </muws2:Capability>
```

1326 4.1.5 Events

1327 To support notifications on a change in a relationship, the following notification topics are defined
1328 in the relationships capability:

```
1329 <wstop:Topic name="RelationshipCreated"  
1330 messageTypes="muws2:RelationshipCreatedNotification">  
1331   <wstop:MessagePattern  
1332     Dialect="http://www.w3.org/TR/1999/REC-xpath-  
1333     19991116">//muws1:ManagementEvent[count(muws2:RelationshipCreatedNotificati  
1334     on)=1]  
1335   </wstop:MessagePattern>  
1336 </wstop:Topic>  
1337 <wstop:Topic name="RelationshipDeleted"  
1338 messageTypes="muws2:RelationshipDeletedNotification">  
1339   <wstop:MessagePattern  
1340     Dialect="http://www.w3.org/TR/1999/REC-xpath-  
1341     19991116">//muws1:ManagementEvent[count(muws2:RelationshipDeletedNotificati  
1342     on)=1]  
1343   </wstop:MessagePattern>  
1344 </wstop:Topic>
```

1345 **muws2:RelationshipCreated** indicates the addition of a new relationship. It is RECOMMENDED
1346 that a consumer subscribe to this notification with an appropriate selector against the content of
1347 notification messages in order to reduce the volume of received messages. Each notification
1348 message contains at least the following information:

```
1349 <RelationshipCreatedNotification>
1350   <Relationship/>
1351 </RelationshipCreatedNotification>
```

1352 **muws:RelationshipDeleted** indicates removal of an existing relationship. It is
1353 RECOMMENDED that a consumer subscribe to this notification with an appropriate selector
1354 against the content of notification messages in order to reduce the volume of received messages.
1355 Each notification message contains at least the following information:

1356

```
1357 <RelationshipDeletedNotification>
1358   <Relationship/>
1359 </RelationshipDeletedNotification>
1360
```

1361 4.2 Relationship Resource Capability

1362 The manageability capability URI for this capability is
1363 <http://docs.oasis-open.org/wsdm/muws/capabilities/RelationshipResource>

1364 4.2.1 Definition

1365 A Web service endpoint, in addition to providing access to a relationship may also represent a
1366 relationship. Representing a relationship means that an endpoint is able to provide relationship
1367 information as described in section 4.1.2.2. In this case, a Web service endpoint MUST be a WS-
1368 Resource, as defined by the WSRF. One such WS-Resource provides information about one
1369 relationship instance. Representing a relationship as a WS-Resource is useful when a
1370 manageability model defines additional properties, operations or events for a relationship.

1371 In order to represent a relationship as a WS-Resource, a set of properties is normatively required.
1372 The rest of the representation depends upon the relationship manageability model and discretion
1373 of the provider of a WS-Resource and relationship.

1374 4.2.2 Properties

1375 The Relationship Resource capability defines the following properties.

```
1376 <muws2:Name>xs:string</muws2:Name> ?
```

1377 **muws2:Name** is an element as defined by the Relationship/Name in section 4.1.2.2. It is
1378 OPTIONAL.

1379

```
1380 <muws2:Type>muws2:RelationshipTypeType</muws2:Type>
```

1381 **muws2:Type** is an element as defined by the Relationship/Type in section 4.1.2.2. It is
1382 REQUIRED and can only appear once.

1383

```
1384 <muws2:Participant>
1385   <muws1:ManageabilityEndpointReference/> *
1386   <wsa:EndpointReference/> *
1387   <muws1:ResourceId/> ?
1388   <muws2:Role>xs:anyURI</muws2:Role>
1389   {any} *
1390 </muws2:Participant>
```

1391

1392 **muws2:Participant** is an element as defined by the Relationship/Participant in section 4.1.2.2.
1393 This element MUST appear at least twice, and exactly once per participant in the relationship.

1394 Even though the optional Self element is still in the Participant element schema, the Self element
1395 should not be specified in participants of relationships when they are resources, as shown in the
1396 example above.

1397 **4.2.3 Events**

1398 The *muwse:RelationshipResourceCapability* topic defined below is used for events related to the
1399 *Relationship Resource* capability.

```
1400 <wstop:Topic name="RelationshipResourceCapability"  
1401           messageTypes="muws1:ManagementEvent">  
1402 </wstop:Topic>
```

1403 **4.3 Advertisement**

1404 The manageability capability URI for the Advertisement capability is
1405 <http://docs.oasis-open.org/wsdm/muws/capabilities/Advertisement>

1406 **4.3.1 Definition**

1407 The *Advertisement* capability is exposed by a Web service that is able to provide a notification on
1408 the creation or the destruction of a manageable resource. Since a consumer cannot register for a
1409 notification on a resource before the resource is created, a creation event is reported for some
1410 other resource by the implementer of a “lifetime notification” capability.

1411 Note that this capability may be implemented by a manageable resource or by some other
1412 service (see section 4 on the distinction between “manageability capability” and “management-
1413 related capability”). A service might offer a capability to notify on the creation or the destruction of
1414 a resource even though the service itself is not manageable. For example, if a system includes a
1415 registry, to which a resource is added as soon as it is created, and from which it is removed when
1416 it is destroyed, then this registry could expose the *Advertisement* capability and use it to share
1417 information about resource creation and destruction events with manageability consumers.
1418 Likewise, a resource factory might emit creation events for a resource it creates, yet the factory
1419 itself might not be manageable. Another example is a container, a J2EE server or a business
1420 process execution engine for example, that can send a notification when a contained resource is
1421 created.

1422 This capability defines four topics used for notification but does not define any property or
1423 operation.

1424 In addition to advertisement by sending notifications, as defined in this capability, another
1425 approach for advertisement is to register a manageable resource in a registry. A resource
1426 advertised in this way can be discovered using the mechanisms introduced in section 5.2.

1427 **4.3.2 Events**

1428 The Advertisement capability defines four notification topics:

```
1429 <wstop:Topic name="ManageabilityEndpointCreation"  
1430 messageTypes="muws2:CreationNotification">  
1431   <wstop:MessagePattern  
1432     Dialect="http://www.w3.org/TR/1999/REC-xpath-  
1433     19991116">//muws1:ManagementEvent[count(muws2:CreationNotification)=1]  
1434   </wstop:MessagePattern>  
1435   <wstop:Topic name="ManageableResourceCreation"  
1436 messageTypes="muws2:CreationNotification">  
1437     <wstop:MessagePattern  
1438       Dialect="http://www.w3.org/TR/1999/REC-xpath-  
1439       19991116">//muws1:ManagementEvent[count(muws2:CreationNotification)=1]  
1440     </wstop:MessagePattern>
```

```

1441     </wstop:Topic>
1442 </wstop:Topic>
1443     <wstop:Topic name="ManageabilityEndpointDestruction"
1444 messageTypes="muws2:DestructionNotification">
1445     <wstop:MessagePattern
1446       Dialect="http://www.w3.org/TR/1999/REC-xpath-
1447 19991116">//muws1:ManagementEvent[count(muws2:DestructionNotification)=1]
1448     </wstop:MessagePattern>
1449     <wstop:Topic name="ManageableResourceDestruction"
1450 messageTypes="muws2:DestructionNotification"/>
1451     <wstop:MessagePattern
1452       Dialect="http://www.w3.org/TR/1999/REC-xpath-
1453 19991116">//muws1:ManagementEvent[count(muws2:DestructionNotification)=1]
1454     </wstop:MessagePattern>
1455   </wstop:Topic>
1456 </wstop:Topic>

```

1457 The **"muws2:ManageabilityEndpointCreation"** topic corresponds to notification on the creation
1458 of a new manageability endpoint for a new or existing resource. A manageability endpoint may be
1459 created in conjunction with, or independent of, the creation of the manageable resource. A new
1460 manageability endpoint could be the first one for a resource or be an addition to others. An
1461 associated *muws2:CreationNotification* message contains the EPR of a newly created
1462 manageability endpoint.

1463 The **"muws2:ManageableResourceCreation"** topic is a specialization of the "Manageability
1464 EndpointCreation" topic. This topic corresponds to the case where a resource itself is newly
1465 created. Note that if a resource is created that is not manageable (i.e. which does not have a
1466 manageability endpoint) no notification on this topic will be sent. If a resource and a
1467 manageability endpoint for the resource are created then a notification will be sent to a subscriber
1468 on this topic.

1469 The **"muws2:ManageabilityEndpointDestruction"** topic corresponds to notification on the
1470 destruction of a manageability endpoint. It does not imply that the associated resource was
1471 destroyed. An associated *muws2:DestructionNotification* message contains the
1472 *muws2:ResourceId* that a newly destroyed manageability endpoint provided for the resource
1473 before its destruction.

1474 The **"muws2:ManageableResourceDestruction"** topic is a specialization of the
1475 "ManageabilityEndpointDestruction" topic. This topic corresponds to the case where a resource
1476 itself is destroyed at the same time as the manageability endpoint. Note that if a resource is
1477 destroyed that is not manageable (i.e. which does not have a manageability endpoint) no
1478 notification on this topic will be sent. An associated *muws2:DestructionNotification* message
1479 contains the *muws2:ResourceId* that a newly destroyed manageability endpoint provided for the
1480 resource before its destruction.

1481 The content element for these topics are described as follows:

```

1482 <muws2:CreationNotification">
1483   <muws1:ManageabilityEndpointReference"/> *
1484 </muws2:CreationNotification">

```

1485 **muws2:CreationNotification/muws1:ManageabilityEndpointReference** is a reference to the
1486 manageability endpoint of a newly created resource. There can be more than one such reference
1487 if there is more than one known manageability endpoint.

1488

```

1489 <muws2:DestructionNotification">
1490   <muws1:ResourceId"/> ?
1491 </muws2:DestructionNotification">

```

1492 **muws2:DestructionNotification/muws1:ResourceId** is the *ResourceId* of a newly destroyed
1493 resource.

5 Discovery

1495

1496 Many forms of discovery are supported by Web services. This specification does not prescribe a
1497 normative method for discovering manageability services. It is expected that discovery methods
1498 commonly used for Web services will be used as discovery methods for manageability services.
1499 The goal of discovery is to obtain the EPR of a manageability endpoint. The Advertisement
1500 capability (section 4.3), when supported, provides one way to facilitate discovery via events. This
1501 section also describes two other ways to discover manageable resources. These are just some of
1502 the discovery methods that can be used.

1503 The only normative requirement relative to discovering manageability services is that a
1504 manageability service **MUST** provide the Identity capability as defined by MUWS. As a result of
1505 this requirement, a consumer can inspect the WSDL description for a Web service or attempt to
1506 use the Identity capability of a Web service to determine if a discovered service acts as a
1507 manageability service. If a discovered service provides at least the Identity capability as defined
1508 by MUWS, then it is a manageability service.

5.1 Discovery using Relationships

1509 There are at least two scenarios in which a relationship can be used to discover a manageable
1510 resource.
1511

1512 The first scenario is when a manageable resource points to some other manageable resource
1513 through a relationship. A manageable resource that supports the Relationship capability enables
1514 discovery of an EPR for some other resource that participates in a relationship with the
1515 manageable resource. This is done by using the "Relationship" property defined in section 4.1.3
1516 or invoking the operations defined in section 4.1.4. Any EPRs contained in such a response
1517 message may be used by the manageability consumer to disambiguate a manageable resource
1518 in an exchange of messages with a manageability endpoint.

1519 The second scenario is when a consumer has access to a WS-Resource representing a
1520 relationship and the relationship has a manageable resource as a member. A consumer can then
1521 use the properties of the Relationship Resource capability to retrieve any EPRs of a manageable
1522 resource participating in the relationship.

5.2 Discovery using Registries

1523 In addition to emitting a notification on the creation and the destruction of a resource as defined
1524 by the Advertisement capability in section 4.3, a resource can be advertised to a registry by
1525 invoking an insertion interface of the registry. A consumer can then discover a manageable
1526 resource by invoking a query interface of the registry.
1527

1528 The WSRF WS-Service Group specification **[WS-SG]** defines a type of registry, along with the
1529 message exchanges used to interact with a registry of this type. It is **RECOMMENDED** that a
1530 registry used to discover a manageable resource conforms to the WS-Service Group specification
1531 and that the registry conform to the following additional constraints:

1532 The service group **SHOULD** include as properties the following two elements:

```
1533 <wssg:MembershipContentRule  
1534   MemberInterface="muws1:Identity"  
1535   ContentElements="muws1:ResourceId">  
1536 <wssg:MembershipContentRule  
1537   MemberInterface="muws1:ManageabilityCharacteristics"  
1538   ContentElements="muws1:ManageabilityCapability">
```

1539 The service group **MAY** also have any other "MembershipContentRule", including a rule with an
1540 empty value for both MemberInterface and ContentElements. In effect, this lifts any constraint on

1541 a member of the service group. The two membership content rules defined above are useful even
1542 in a service group with no effective constraint because they allow querying the service group on
1543 the “ResourceId” and “ManageabilityCapability” properties.

1544 When adding a manageability endpoint for a resource to the membership of a service group using
1545 the “Add” operation, the requestor SHOULD include the *muws1:ResourceId* element of a
1546 manageable resource in a *wssg:Add/wssg:Content* element of a request, even if the service
1547 group supports additional membership content rules that would have permitted registration of a
1548 manageability endpoint in the service group without providing this content element. Similarly, if
1549 the manageable resource supports the Manageability Characteristics capability, then the
1550 consumer SHOULD include all the *muws1:ManageabilityCapability* elements of a manageable
1551 resource in a *wssg:Add/wssg:Content* element of a request, even if the service group supports
1552 additional membership content rules that would have permitted registration of the manageability
1553 endpoint in the service group without providing this content element.

1554 Like any manageability endpoint, a manageability endpoint listed in a resource registry MUST
1555 implement the Identity capability defined in **[MUWS Part 1]**. In addition, in order to facilitate
1556 discovery, the manageability endpoint SHOULD implement the Manageability Characteristics
1557 capability as defined in **[MUWS Part 1]**.

1558

6 References

1559

6.1 Normative

1560

1561 **[MUWS Part 1]** Vaughn Bullard, *Web Services Distributed Management: Management*
1562 *using Web Services (MUWS 1.1) Part 1*, OASIS Committee Draft, March
1563 2006, <http://docs.oasis-open.org/wsdm/wsdm-muws1-1.1-spec-os-01.pdf>

1564

1565 **[XML1.0 3rd Edition]**
1566 Tim Bray, et al., *Extensible Markup Language (XML) 1.0 (Third Edition)*,
1567 W3C Recommendation, February 2004, <http://www.w3.org/TR/REC-xml>

1568

1569 **[XML Schema Part 1]**
1570 Henry S. Thompson, et al. *XML Schema Part 1: Structures*, W3C
1571 Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-1/>

1572

1573 **[XML Schema Part 2]**
1574 Paul V. Biron, et al. *XML Schema Part 2: Datatypes*, W3C
1575 Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-2/>

1576

1577 **[WSDL]** Erik Christensen, et al., *Web services Description Language (WSDL)*
1578 *1.1*, W3C Note, March 2001, <http://www.w3.org/TR/wsdl>

1579

1580 **[WS-Resource]** Steve Graham, et al. *Web Service Resource 1.2 (WS-Resource)*, OASIS
1581 Standard, April 2006, http://docs.oasis-open.org/wsr/wsr/ws_resource-1.2-spec-os.pdf

1582

1584 **[WS-Addressing]** Don Box, et al., *Web services Addressing (WS-Addressing)*, W3C
1585 Member Submission, August 2004, <http://www.w3.org/TR/ws-addr-core>

1586

1587 **[WS-RP]** Steve Graham, et al., *Web Services Resource Properties 1.2 (WS-
1588 ResourceProperties)*, OASIS Standard, April 2006, [http://docs.oasis-
1589 open.org/wsr/wsr/ws_resource_properties-1.2-spec-os.pdf](http://docs.oasis-open.org/wsr/wsr/ws_resource_properties-1.2-spec-os.pdf)

1590

1591 **[XPath 1.0]** James Clark, et al., *XML Path Language (XPath) Version 1.0*, W3C
1592 Recommendation, November 1999, [http://www.w3.org/TR/1999/REC-
1593 xpath-19991116](http://www.w3.org/TR/1999/REC-xpath-19991116)

1594

1595 **[WSN]** Steve Graham, et al., *Web Services Base Notification 1.2 (WS-
1596 BaseNotification)*, OASIS Public Review Draft, June 2006,
1597 [http://www.oasis-open.org/apps/org/workgroup/wsn/wsn-
1598 ws_base_notification-1.3-spec-pr-03.pdf](http://www.oasis-open.org/apps/org/workgroup/wsn/wsn-ws_base_notification-1.3-spec-pr-03.pdf)

1599

1600 **[WST]** William Vambenepe, *Web Services Topics 1.2 (WS-Topics)*, OASIS
1601 Public Review Draft, June 2006, [http://www.oasis-
1602 open.org/apps/org/workgroup/wsn/wsn-ws_topics-1.3-spec-pr-02.pdf](http://www.oasis-open.org/apps/org/workgroup/wsn/wsn-ws_topics-1.3-spec-pr-02.pdf)

1603

1604 **[RFC3066]** IETF (Internet Engineering Task Force). RFC 3066: *Tags for the*
1605 *Identification of Languages*, ed. H. Alvestrand. 2001,
1606 <http://www.ietf.org/rfc/rfc3066.txt>
1607

1608 **[WS-RL]** Latha Srinivasan, et al., *Web Services Resource Lifetime 1.2 (WS-*
1609 *ResourceLifetime)*, OASIS Standard, April 2006, [http://docs.oasis-](http://docs.oasis-open.org/wsrf/wsrf-ws_resource_lifetime-1.2-spec-os.pdf)
1610 [open.org/wsrf/wsrf-ws_resource_lifetime-1.2-spec-os.pdf](http://docs.oasis-open.org/wsrf/wsrf-ws_resource_lifetime-1.2-spec-os.pdf)
1611

1612 **[WS-SG]** Tom Maguire, et al., *Web Services Service Group 1.2 (WS-*
1613 *ServiceGroup)*, OASIS Standard, April 2006, [http://docs.oasis-](http://docs.oasis-open.org/wsrf/wsrf-ws_service_group-1.2-spec-os.pdf)
1614 [open.org/wsrf/wsrf-ws_service_group-1.2-spec-os.pdf](http://docs.oasis-open.org/wsrf/wsrf-ws_service_group-1.2-spec-os.pdf)
1615

1616 **6.2 Non-normative**

1617 **[MOWS]** Kirk Wilson, *Web Services Distributed Management: Management of*
1618 *Web Services (WSDM-MOWS) 1.1*, OASIS Committee Draft, February
1619 2006, <http://docs.oasis-open.org/wsdm/wsdm-mows-1.1-spec-os-01.pdf>
1620

1621 **[SOAP]** Don Box, et al., *Simple Object Access Protocol (SOAP) 1.1*, W3C Note,
1622 May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
1623

1624 **[WS-RF]** WSRF OASIS technical committee, [http://www.oasis-](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf)
1625 [open.org/committees/tc_home.php?wg_abbrev=wsrf](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf)
1626

1627 **[WSRMD]** Steve Graham, et al., *Web Services Resource Metadata 1.0 (WS-*
1628 *ResourceMetadataDescriptor)*, OASIS Working Draft, January 2006,
1629 [http://www.oasis-open.org/wsrf/wsrf-ws_resource_metadata_descriptor-](http://www.oasis-open.org/wsrf/wsrf-ws_resource_metadata_descriptor-1.0-spec-06.pdf)
1630 [1.0-spec-06.pdf](http://www.oasis-open.org/wsrf/wsrf-ws_resource_metadata_descriptor-1.0-spec-06.pdf)
1631

1632 **Appendix A. Acknowledgements**

1633 WSDM Management Using Web Services Part 2 Version 1.0 Acknowledgements

1634 The following individuals were members of the committee when the WSDM MUWS Version 1.0
1635 was approved by the technical committee

1636 Guru Bhat, Jeff Bohren, Winston Bumpus, Nick Butler, Brian Carroll, Fred Carter, Michael
1637 Clements, David Cox, John DeCarlo, Andreas Dharmawan, Mark Ellison, John Fuller, Paul
1638 Lipton, Heather Kreger, Hal Lockhart, Frederico Maciel, Tom Maguire, Bryan Murray, Richard
1639 Nikula, Mark Peel, Richard Pelavin, Homayoun Pourheidari, Warren Roberts, Karl Schopmeyer,
1640 Igor Sedukhin, David Snelling, Thomas Studwell, William Vambenepe, Andrea Westerinen, Jim
1641 Willits, Zhili Zhang.

1642 In addition, the following non-member employees of member companies made contribution to the
1643 specification: Maryann Hondo, Ian Springer, John Gerken, David Melgar, Mitsunori Satomi.

1644

1645 WSDM Management Using Web Services Part 2 Version 1.1 Acknowledgements

1646 The following people made contributions to the WSDM MUWS Version 1.1 specification: Vaughn
1647 Bullard, Fred Carter, David Cox, Zulah Eckert, Mark Ellison, Heather Kreger, Frederico Maciel,
1648 Bryan Murray, Richard Nikula, Mitsunori Satomi, Thomas Studwell, Kirk Wilson, Zhili Zhang with
1649 special thanks to Vaughn Bullard and Mark Ellison as editors.

1650 The following individuals were members of the committee while the WSDM MUWS Version 1.1
1651 specification was developed and approved by the technical committee: Guru Bhat, Jeff Bohren,
1652 Vaughn Bullard, Winston Bumpus, Fred Carter, Michael Clements, David Cox, Zulah Eckert,
1653 Mark Ellison, John Fuller, Tony Gullato, Heather Kreger, Richard Landau, Frederico Maciel, Tom
1654 Maguire, David Melgar, Bryan Murray, Richard Nikula, Mark Peel, Mitsunori Satomi, Thomas
1655 Studwell, William Vambenepe, Kirk Wilson, Zhili Zhang.

Appendix B. Notices

1656

1657 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
1658 that might be claimed to pertain to the implementation or use of the technology described in this
1659 document or the extent to which any license under such rights might or might not be available;
1660 neither does it represent that it has made any effort to identify any such rights. Information on
1661 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
1662 website. Copies of claims of rights made available for publication and any assurances of licenses
1663 to be made available, or the result of an attempt made to obtain a general license or permission
1664 for the use of such proprietary rights by implementers or users of this specification, can be
1665 obtained from the OASIS Executive Director.

1666 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
1667 applications, or other proprietary rights which may cover technology that may be required to
1668 implement this specification. Please address the information to the OASIS Executive Director.

1669 Copyright © OASIS Open 2003-2006. *All Rights Reserved.*

1670 This document and translations of it may be copied and furnished to others, and derivative works
1671 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
1672 published and distributed, in whole or in part, without restriction of any kind, provided that the
1673 above copyright notice and this paragraph are included on all such copies and derivative works.
1674 However, this document itself does not be modified in any way, such as by removing the
1675 copyright notice or references to OASIS, except as needed for the purpose of developing OASIS
1676 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
1677 Property Rights document must be followed, or as required to translate it into languages other
1678 than English.

1679 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
1680 successors or assigns.

1681 This document and the information contained herein is provided on an "AS IS" basis and OASIS
1682 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
1683 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
1684 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
1685 PARTICULAR PURPOSE.

1686

1687

1688

1689

1690

Appendix C. Schemas

1691

```
1692 <?xml version="1.0" encoding="utf-8" ?>
1693 <xs:schema targetNamespace="http://docs.oasis-open.org/wsdm/muws2-2.xsd"
1694 xmlns:muws2="http://docs.oasis-open.org/wsdm/muws2-2.xsd"
1695     xmlns:muws1="http://docs.oasis-open.org/wsdm/muws1-2.xsd"
1696     xmlns:wsa="http://www.w3.org/2005/08/addressing"
1697     xmlns:xs="http://www.w3.org/2001/XMLSchema"
1698     elementFormDefault="qualified" attributeFormDefault="unqualified">
1699     <xs:import namespace="http://docs.oasis-open.org/wsdm/muws1-2.xsd"
1700     schemaLocation="http://docs.oasis-open.org/wsdm/muws1-2.xsd" />
1701     <xs:import namespace="http://www.w3.org/2005/08/addressing"
1702     schemaLocation="http://www.w3.org/2005/08/addressing/ws-addr.xsd" />
1703     <xs:complexType name="LangString">
1704         <xs:simpleContent>
1705             <xs:extension base="xs:string">
1706                 <xs:attribute ref="xml:lang" use="required" />
1707                 <xs:anyAttribute namespace="##other" />
1708             </xs:extension>
1709         </xs:simpleContent>
1710     </xs:complexType>
1711     <!-- Begin properties for the Description capability -->
1712     <xs:element name="Caption" type="muws2:LangString" />
1713     <xs:element name="Description" type="muws2:LangString" />
1714     <xs:element name="Version" type="xs:string" />
1715     <!-- End properties for the Description capability -->
1716     <xs:complexType name="CategoryType">
1717         <xs:sequence>
1718             <xs:any minOccurs="0" namespace="##any"
1719     processContents="lax" />
1720         </xs:sequence>
1721     </xs:complexType>
1722     <xs:complexType name="StateType">
1723         <xs:complexContent>
1724             <xs:extension base="muws2:CategoryType" />
1725         </xs:complexContent>
1726     </xs:complexType>
1727     <xs:element name="State" type="muws2:StateType" />
1728     <xs:element name="EnteredState" type="muws2:StateType" />
1729     <xs:element name="PreviousState" type="muws2:StateType" />
1730     <xs:complexType name="StateTransitionType">
1731         <xs:sequence>
1732             <xs:element ref="muws2:EnteredState" />
1733             <xs:element ref="muws2:PreviousState" minOccurs="0" />
1734             <xs:any minOccurs="0" maxOccurs="unbounded"
1735     namespace="##other" processContents="lax" />
1736         </xs:sequence>
1737         <xs:attribute name="TransitionIdentifier" type="xs:anyURI"
1738     use="optional" />
1739         <xs:attribute name="Time" type="xs:dateTime" use="required" />
1740         <xs:anyAttribute namespace="##other" />
1741     </xs:complexType>
1742     <xs:element name="StateTransition" type="muws2:StateTransitionType" />
1743     <!-- Begin properties for the OperationalStatus capability -->
1744     <xs:element name="OperationalStatus">
1745         <xs:simpleType>
1746             <xs:restriction base="xs:string">
1747                 <xs:enumeration value="Available" />
1748                 <xs:enumeration value="PartiallyAvailable" />
1749                 <xs:enumeration value="Unavailable" />
1750                 <xs:enumeration value="Unknown" />

```

```

1751         </xs:restriction>
1752     </xs:simpleType>
1753 </xs:element>
1754 <!-- End properties for the OperationalStatus capability -->
1755 <xs:attributeGroup name="MetricAttributes">
1756     <xs:attribute name="ResetAt" type="xs:dateTime" />
1757     <xs:attribute name="LastUpdated" type="xs:dateTime" />
1758     <xs:attribute name="Duration" type="xs:duration" />
1759 </xs:attributeGroup>
1760 <!-- Begin properties for the Metrics capability -->
1761 <xs:element name="CurrentTime" type="xs:dateTime" />
1762 <!-- End properties for the Metrics capability -->
1763 <xs:complexType name="RelationshipTypeType">
1764     <xs:complexContent>
1765         <xs:extension base="muws2:CategoryType" />
1766     </xs:complexContent>
1767 </xs:complexType>
1768 <xs:element name="Self">
1769     <xs:complexType />
1770 </xs:element>
1771 <xs:complexType name="RelationshipParticipantType">
1772     <xs:sequence>
1773         <xs:element ref="muws2:Self" minOccurs="0" />
1774         <xs:element ref="muws1:ManageabilityEndpointReference"
1775 minOccurs="0" maxOccurs="unbounded" />
1776         <xs:element ref="wsa:EndpointReference" minOccurs="0"
1777 maxOccurs="unbounded" />
1778         <xs:element ref="muws1:ResourceId" minOccurs="0" />
1779         <xs:element name="Role" type="xs:anyURI" />
1780         <xs:any minOccurs="0" maxOccurs="unbounded"
1781 namespace="##other" processContents="lax" />
1782     </xs:sequence>
1783     <xs:anyAttribute namespace="##other" />
1784 </xs:complexType>
1785 <!-- Begin properties for the RelationshipResource capability -->
1786 <xs:element name="Name" type="xs:string" />
1787 <xs:element name="Type" type="muws2:RelationshipTypeType" />
1788 <xs:element name="Participant" type="muws2:RelationshipParticipantType"
1789 />
1790 <!-- End properties for the RelationshipResource capability -->
1791 <xs:complexType name="RelationshipType">
1792     <xs:sequence>
1793         <xs:element ref="muws2:Name" minOccurs="0" />
1794         <xs:element ref="muws2:Type" />
1795         <xs:element ref="muws2:Participant" minOccurs="2"
1796 maxOccurs="unbounded" />
1797         <xs:element name="AccessEndpointReference"
1798 type="wsa:EndpointReferenceType" minOccurs="0" />
1799         <xs:any minOccurs="0" maxOccurs="unbounded"
1800 namespace="##other" processContents="lax" />
1801     </xs:sequence>
1802     <xs:anyAttribute namespace="##other" />
1803 </xs:complexType>
1804 <!-- Begin properties for the Relationship capability -->
1805 <xs:element name="Relationship" type="muws2:RelationshipType" />
1806 <!-- End properties for the Relationship capability -->
1807 <xs:element name="RelationshipCreatedNotification">
1808     <xs:complexType>
1809         <xs:sequence>
1810             <xs:element ref="muws2:Relationship" />
1811             <xs:any minOccurs="0" maxOccurs="unbounded"
1812 namespace="##other" processContents="lax" />
1813         </xs:sequence>

```

```

1814         <xs:anyAttribute namespace="##other" />
1815     </xs:complexType>
1816 </xs:element>
1817 <xs:element name="RelationshipDeletedNotification">
1818     <xs:complexType>
1819         <xs:sequence>
1820             <xs:element ref="muws2:Relationship" />
1821             <xs:any minOccurs="0" maxOccurs="unbounded"
1822 namespace="##other" processContents="lax" />
1823         </xs:sequence>
1824         <xs:anyAttribute namespace="##other" />
1825     </xs:complexType>
1826 </xs:element>
1827 <xs:element name="QueryRelationshipsByType">
1828     <xs:complexType>
1829         <xs:sequence>
1830             <xs:element name="RequestedType" type="xs:QName" />
1831         </xs:sequence>
1832     </xs:complexType>
1833 </xs:element>
1834 <xs:element name="QueryRelationshipsByTypeResponse">
1835     <xs:complexType>
1836         <xs:sequence>
1837             <xs:element ref="muws2:Relationship" minOccurs="0"
1838 maxOccurs="unbounded" />
1839         </xs:sequence>
1840     </xs:complexType>
1841 </xs:element>
1842 <xs:element name="CreationNotification">
1843     <xs:complexType>
1844         <xs:sequence>
1845             <xs:element
1846 ref="muws1:ManageabilityEndpointReference" minOccurs="0" maxOccurs="unbounded"
1847 />
1848         </xs:sequence>
1849         <xs:anyAttribute namespace="##other" />
1850     </xs:complexType>
1851 </xs:element>
1852 <xs:element name="DestructionNotification">
1853     <xs:complexType>
1854         <xs:sequence>
1855             <xs:element ref="muws1:ResourceId" minOccurs="0" />
1856         </xs:sequence>
1857         <xs:anyAttribute namespace="##other" />
1858     </xs:complexType>
1859 </xs:element>
1860 <xs:complexType name="SituationCategoryType">
1861     <xs:complexContent>
1862         <xs:extension base="muws2:CategoryType" />
1863     </xs:complexContent>
1864 </xs:complexType>
1865 <xs:complexType name="SubstitutableMsgType">
1866     <xs:sequence>
1867         <xs:element name="Value" type="xs:anySimpleType"
1868 minOccurs="0" maxOccurs="unbounded" />
1869     </xs:sequence>
1870     <xs:attribute name="MsgId" type="xs:string" use="required" />
1871     <xs:attribute name="MsgIdType" type="xs:anyURI" use="required" />
1872 </xs:complexType>
1873 <xs:complexType name="SituationType">
1874     <xs:sequence>
1875         <xs:element name="SituationCategory"
1876 type="muws2:SituationCategoryType" />

```

```

1877         <xs:element name="SuccessDisposition" minOccurs="0">
1878             <xs:simpleType>
1879                 <xs:restriction base="xs:string">
1880                     <xs:enumeration value="Successful" />
1881                     <xs:enumeration value="Unsuccessful" />
1882                 </xs:restriction>
1883             </xs:simpleType>
1884         </xs:element>
1885         <xs:element name="SituationTime" type="xs:dateTime" />
1886         <xs:element name="Priority" type="xs:short" minOccurs="0"
1887 />
1888         <xs:element name="Severity" type="xs:short" minOccurs="0"
1889 />
1890         <xs:element name="Message" type="muws2:LangString"
1891 minOccurs="0" />
1892         <xs:element name="SubstitutableMsg"
1893 type="muws2:SubstitutableMsgType" minOccurs="0" />
1894     </xs:sequence>
1895 </xs:complexType>
1896 <xs:element name="Situation" type="muws2:SituationType" />
1897 <xs:complexType name="EventCorrelationPropertiesType">
1898     <xs:sequence>
1899         <xs:element name="repeatCount" minOccurs="0" maxOccurs="1">
1900             <xs:simpleType>
1901                 <xs:restriction base="xs:short">
1902                     <xs:minInclusive value="0" />
1903                 </xs:restriction>
1904             </xs:simpleType>
1905         </xs:element>
1906         <xs:element name="elapsedTime" minOccurs="0" maxOccurs="1">
1907             <xs:simpleType>
1908                 <xs:restriction base="xs:long">
1909                     <xs:minInclusive value="0" />
1910                 </xs:restriction>
1911             </xs:simpleType>
1912         </xs:element>
1913     </xs:sequence>
1914     <xs:attribute name="sequenceNumber" type="xs:unsignedLong"/>
1915 </xs:complexType>
1916 <xs:element name="EventCorrelationProperties"
1917 type="muws2:EventCorrelationPropertiesType" />
1918 <xs:complexType name="MsgCatalogInformationType">
1919     <xs:sequence>
1920         <xs:element name="msgCatalog" type="xs:anyURI"
1921 minOccurs="1" />
1922         <xs:element name="msgCatalogType" type="xs:anyURI"
1923 minOccurs="0" />
1924     </xs:sequence>
1925 </xs:complexType>
1926 <xs:element name="MsgCatalogInformation"
1927 type="muws2:MsgCatalogInformationType" />
1928 <!-- ##### Metadata description elements ##### -->
1929 <xs:element name="Capability" type="xs:anyURI" />
1930 <xs:complexType name="DialectableExpressionType" mixed="true">
1931     <xs:sequence>
1932         <xs:any namespace="##other" processContents="lax"
1933 minOccurs="0" maxOccurs="unbounded" />
1934     </xs:sequence>
1935     <xs:attribute name="Dialect" type="xs:anyURI" use="required" />
1936     <xs:anyAttribute namespace="##other" />
1937 </xs:complexType>
1938 <xs:element name="ValidWhile" type="muws2:DialectableExpressionType" />
1939 <xs:element name="Units" type="xs:string" />

```

```

1940 <xs:element name="ChangeType">
1941 <xs:simpleType>
1942 <xs:restriction base="xs:string">
1943 <xs:enumeration value="Counter" />
1944 <xs:enumeration value="Gauge" />
1945 <xs:enumeration value="Unknown" />
1946 </xs:restriction>
1947 </xs:simpleType>
1948 </xs:element>
1949 <xs:element name="TimeScope">
1950 <xs:simpleType>
1951 <xs:restriction base="xs:string">
1952 <xs:enumeration value="Interval" />
1953 <xs:enumeration value="PointInTime" />
1954 <xs:enumeration value="SinceReset" />
1955 </xs:restriction>
1956 </xs:simpleType>
1957 </xs:element>
1958 <xs:element name="GatheringTime">
1959 <xs:simpleType>
1960 <xs:restriction base="xs:string">
1961 <xs:enumeration value="OnChange" />
1962 <xs:enumeration value="Periodic" />
1963 <xs:enumeration value="OnDemand" />
1964 <xs:enumeration value="Unknown" />
1965 </xs:restriction>
1966 </xs:simpleType>
1967 </xs:element>
1968 <xs:element name="CalculationInterval" type="xs:duration" />
1969 <xs:element name="MetricGroup" type="xs:anyURI" />
1970 <xs:element name="PostCondition" type="muws2:DialectableExpressionType"
1971 />
1972 <!-- ===== StartSituation ===== -->
1973 <xs:element name="StartSituation">
1974 <xs:complexType>
1975 <xs:complexContent>
1976 <xs:restriction base="muws2:SituationCategoryType" />
1977 </xs:complexContent>
1978 </xs:complexType>
1979 </xs:element>
1980 <xs:element name="StartInitiated">
1981 <xs:complexType>
1982 <xs:complexContent>
1983 <xs:restriction base="muws2:SituationCategoryType">
1984 <xs:sequence>
1985 <xs:element ref="muws2:StartSituation"
1986 />
1987 </xs:sequence>
1988 </xs:restriction>
1989 </xs:complexContent>
1990 </xs:complexType>
1991 </xs:element>
1992 <xs:element name="RestartInitiated">
1993 <xs:complexType>
1994 <xs:complexContent>
1995 <xs:restriction base="muws2:SituationCategoryType">
1996 <xs:sequence>
1997 <xs:element ref="muws2:StartSituation"
1998 />
1999 </xs:sequence>
2000 </xs:restriction>
2001 </xs:complexContent>
2002 </xs:complexType>

```

```

2003     </xs:element>
2004     <xs:element name="StartCompleted">
2005         <xs:complexType>
2006             <xs:complexContent>
2007                 <xs:restriction base="muws2:SituationCategoryType">
2008                     <xs:sequence>
2009                         <xs:element ref="muws2:StartSituation"
2010 />
2011                             </xs:sequence>
2012                             </xs:restriction>
2013                         </xs:complexContent>
2014                     </xs:complexType>
2015 </xs:element>
2016 <!-- ===== StopSituation ===== -->
2017 <xs:element name="StopSituation">
2018     <xs:complexType>
2019         <xs:complexContent>
2020             <xs:restriction base="muws2:SituationCategoryType" />
2021         </xs:complexContent>
2022     </xs:complexType>
2023 </xs:element>
2024 <xs:element name="StopInitiated">
2025     <xs:complexType>
2026         <xs:complexContent>
2027             <xs:restriction base="muws2:SituationCategoryType">
2028                 <xs:sequence>
2029                     <xs:element ref="muws2:StopSituation"
2030 />
2031                         </xs:sequence>
2032                         </xs:restriction>
2033                     </xs:complexContent>
2034                 </xs:complexType>
2035 </xs:element>
2036 <xs:element name="AbortInitiated">
2037     <xs:complexType>
2038         <xs:complexContent>
2039             <xs:restriction base="muws2:SituationCategoryType">
2040                 <xs:sequence>
2041                     <xs:element ref="muws2:StopSituation"
2042 />
2043                         </xs:sequence>
2044                         </xs:restriction>
2045                     </xs:complexContent>
2046                 </xs:complexType>
2047 </xs:element>
2048 <xs:element name="PauseInitiated">
2049     <xs:complexType>
2050         <xs:complexContent>
2051             <xs:restriction base="muws2:SituationCategoryType">
2052                 <xs:sequence>
2053                     <xs:element ref="muws2:StopSituation"
2054 />
2055                         </xs:sequence>
2056                         </xs:restriction>
2057                     </xs:complexContent>
2058                 </xs:complexType>
2059 </xs:element>
2060 <xs:element name="StopCompleted">
2061     <xs:complexType>
2062         <xs:complexContent>
2063             <xs:restriction base="muws2:SituationCategoryType">
2064                 <xs:sequence>

```

```

2065         <xs:element ref="muws2:StopSituation"
2066     />
2067         </xs:sequence>
2068     </xs:restriction>
2069 </xs:complexContent>
2070 </xs:complexType>
2071 </xs:element>
2072 <!-- ===== RequestSituation ===== -->
2073 <xs:element name="RequestSituation">
2074     <xs:complexType>
2075         <xs:complexContent>
2076             <xs:restriction base="muws2:SituationCategoryType" />
2077         </xs:complexContent>
2078     </xs:complexType>
2079 </xs:element>
2080 <xs:element name="RequestInitiated">
2081     <xs:complexType>
2082         <xs:complexContent>
2083             <xs:restriction base="muws2:SituationCategoryType">
2084                 <xs:sequence>
2085                     <xs:element
2086 ref="muws2:RequestSituation" />
2087                 </xs:sequence>
2088             </xs:restriction>
2089 </xs:complexContent>
2090 </xs:complexType>
2091 </xs:element>
2092 <xs:element name="RequestCompleted">
2093     <xs:complexType>
2094         <xs:complexContent>
2095             <xs:restriction base="muws2:SituationCategoryType">
2096                 <xs:sequence>
2097                     <xs:element
2098 ref="muws2:RequestSituation" />
2099                 </xs:sequence>
2100             </xs:restriction>
2101 </xs:complexContent>
2102 </xs:complexType>
2103 </xs:element>
2104 <!-- ===== DestroySituation ===== -->
2105 <xs:element name="DestroySituation">
2106     <xs:complexType>
2107         <xs:complexContent>
2108             <xs:restriction base="muws2:SituationCategoryType" />
2109         </xs:complexContent>
2110     </xs:complexType>
2111 </xs:element>
2112 <xs:element name="DestroyInitiated">
2113     <xs:complexType>
2114         <xs:complexContent>
2115             <xs:restriction base="muws2:SituationCategoryType">
2116                 <xs:sequence>
2117                     <xs:element
2118 ref="muws2:DestroySituation" />
2119                 </xs:sequence>
2120             </xs:restriction>
2121 </xs:complexContent>
2122 </xs:complexType>
2123 </xs:element>
2124 <xs:element name="DestroyCompleted">
2125     <xs:complexType>
2126         <xs:complexContent>
2127             <xs:restriction base="muws2:SituationCategoryType">

```

```

2128         <xs:sequence>
2129             <xs:element
2130 ref="muws2:DestroySituation" />
2131         </xs:sequence>
2132     </xs:restriction>
2133 </xs:complexContent>
2134 </xs:complexType>
2135 </xs:element>
2136 <!-- ===== CreateSituation ===== -->
2137 <xs:element name="CreateSituation">
2138     <xs:complexType>
2139         <xs:complexContent>
2140             <xs:restriction base="muws2:SituationCategoryType" />
2141         </xs:complexContent>
2142     </xs:complexType>
2143 </xs:element>
2144 <xs:element name="CreateInitiated">
2145     <xs:complexType>
2146         <xs:complexContent>
2147             <xs:restriction base="muws2:SituationCategoryType">
2148                 <xs:sequence>
2149                     <xs:element ref="muws2:CreateSituation"
2150 />
2151                 </xs:sequence>
2152             </xs:restriction>
2153         </xs:complexContent>
2154     </xs:complexType>
2155 </xs:element>
2156 <xs:element name="CreateCompleted">
2157     <xs:complexType>
2158         <xs:complexContent>
2159             <xs:restriction base="muws2:SituationCategoryType">
2160                 <xs:sequence>
2161                     <xs:element ref="muws2:CreateSituation"
2162 />
2163                 </xs:sequence>
2164             </xs:restriction>
2165         </xs:complexContent>
2166     </xs:complexType>
2167 </xs:element>
2168 <!-- ===== ConnectSituation ===== -->
2169 <xs:element name="ConnectSituation">
2170     <xs:complexType>
2171         <xs:complexContent>
2172             <xs:restriction base="muws2:SituationCategoryType" />
2173         </xs:complexContent>
2174     </xs:complexType>
2175 </xs:element>
2176 <xs:element name="ConnectInitiated">
2177     <xs:complexType>
2178         <xs:complexContent>
2179             <xs:restriction base="muws2:SituationCategoryType">
2180                 <xs:sequence>
2181                     <xs:element
2182 ref="muws2:ConnectSituation" />
2183                 </xs:sequence>
2184             </xs:restriction>
2185         </xs:complexContent>
2186     </xs:complexType>
2187 </xs:element>
2188 <xs:element name="ReconnectInitiated">
2189     <xs:complexType>
2190         <xs:complexContent>

```

```

2191             <xs:restriction base="muws2:SituationCategoryType">
2192                 <xs:sequence>
2193                     <xs:element
2194 ref="muws2:ConnectSituation" />
2195                 </xs:sequence>
2196             </xs:restriction>
2197         </xs:complexContent>
2198     </xs:complexType>
2199 </xs:element>
2200 <xs:element name="ConnectCompleted">
2201     <xs:complexType>
2202     <xs:complexContent>
2203         <xs:restriction base="muws2:SituationCategoryType">
2204             <xs:sequence>
2205                 <xs:element
2206 ref="muws2:ConnectSituation" />
2207             </xs:sequence>
2208         </xs:restriction>
2209     </xs:complexContent>
2210 </xs:complexType>
2211 </xs:element>
2212 <!-- ===== ReportSituation ===== -->
2213 <xs:element name="ReportSituation">
2214     <xs:complexType>
2215     <xs:complexContent>
2216         <xs:restriction base="muws2:SituationCategoryType" />
2217     </xs:complexContent>
2218 </xs:complexType>
2219 </xs:element>
2220 <xs:element name="PerformanceReport">
2221     <xs:complexType>
2222     <xs:complexContent>
2223         <xs:restriction base="muws2:SituationCategoryType">
2224             <xs:sequence>
2225                 <xs:element ref="muws2:ReportSituation"
2226 />
2227             </xs:sequence>
2228         </xs:restriction>
2229     </xs:complexContent>
2230 </xs:complexType>
2231 </xs:element>
2232 <xs:element name="SecurityReport">
2233     <xs:complexType>
2234     <xs:complexContent>
2235         <xs:restriction base="muws2:SituationCategoryType">
2236             <xs:sequence>
2237                 <xs:element ref="muws2:ReportSituation"
2238 />
2239             </xs:sequence>
2240         </xs:restriction>
2241     </xs:complexContent>
2242 </xs:complexType>
2243 </xs:element>
2244 <xs:element name="HeartbeatReport">
2245     <xs:complexType>
2246     <xs:complexContent>
2247         <xs:restriction base="muws2:SituationCategoryType">
2248             <xs:sequence>
2249                 <xs:element ref="muws2:ReportSituation"
2250 />
2251             </xs:sequence>
2252         </xs:restriction>
2253     </xs:complexContent>

```

```

2254         </xs:complexType>
2255     </xs:element>
2256     <xs:element name="StatusReport">
2257         <xs:complexType>
2258             <xs:complexContent>
2259                 <xs:restriction base="muws2:SituationCategoryType">
2260                     <xs:sequence>
2261                         <xs:element ref="muws2:ReportSituation"
2262 />
2263                             </xs:sequence>
2264                         </xs:restriction>
2265                     </xs:complexContent>
2266                 </xs:complexType>
2267             </xs:element>
2268             <xs:element name="TraceReport">
2269                 <xs:complexType>
2270                     <xs:complexContent>
2271                         <xs:restriction base="muws2:SituationCategoryType">
2272                             <xs:sequence>
2273                                 <xs:element ref="muws2:ReportSituation"
2274 />
2275                                     </xs:sequence>
2276                                 </xs:restriction>
2277                             </xs:complexContent>
2278                         </xs:complexType>
2279                     </xs:element>
2280                     <xs:element name="DebugReport">
2281                         <xs:complexType>
2282                             <xs:complexContent>
2283                                 <xs:restriction base="muws2:SituationCategoryType">
2284                                     <xs:sequence>
2285                                         <xs:element ref="muws2:ReportSituation"
2286 />
2287                                             </xs:sequence>
2288                                         </xs:restriction>
2289                                     </xs:complexContent>
2290                                 </xs:complexType>
2291                             </xs:element>
2292                             <xs:element name="LogReport">
2293                                 <xs:complexType>
2294                                     <xs:complexContent>
2295                                         <xs:restriction base="muws2:SituationCategoryType">
2296                                             <xs:sequence>
2297                                                 <xs:element ref="muws2:ReportSituation"
2298 />
2299                                                     </xs:sequence>
2300                                                 </xs:restriction>
2301                                             </xs:complexContent>
2302                                         </xs:complexType>
2303                                     </xs:element>
2304                                 <!-- ===== AvailabilitySituation ===== -->
2305                                 <xs:element name="AvailabilitySituation">
2306                                     <xs:complexType>
2307                                         <xs:complexContent>
2308                                             <xs:restriction base="muws2:SituationCategoryType" />
2309                                         </xs:complexContent>
2310                                     </xs:complexType>
2311                                 </xs:element>
2312                                 <!-- ===== CapabilitySituation ===== -->
2313                                 <xs:element name="CapabilitySituation">
2314                                     <xs:complexType>
2315                                         <xs:complexContent>
2316                                             <xs:restriction base="muws2:SituationCategoryType" />

```

```

2317         </xs:complexContent>
2318     </xs:complexType>
2319 </xs:element>
2320 <!-- ===== ConfigureSituation ===== -->
2321 <xs:element name="ConfigureSituation">
2322     <xs:complexType>
2323         <xs:complexContent>
2324             <xs:restriction base="muws2:SituationCategoryType" />
2325         </xs:complexContent>
2326     </xs:complexType>
2327 </xs:element>
2328 <!-- ===== OtherSituation ===== -->
2329 <xs:element name="OtherSituation">
2330     <xs:complexType>
2331         <xs:complexContent>
2332             <xs:restriction base="muws2:SituationCategoryType" />
2333         </xs:complexContent>
2334     </xs:complexType>
2335 </xs:element>
2336 <!--
2337             SCHEMA COPY Material
2338 Copy and paste element references below into the schema of a resource
2339 properties document.
2340 These references insure that the correct minOccurs/maxOccurs attributes are
2341 specified in a resource property document schema.
2342
2343 NOTE: You must import the MUWS Part 2 schema namespace (MUWS2).
2344
2345     **      Description Properties      **
2346     <xs:element ref="muws2:Caption"
2347         minOccurs="0" maxOccurs="unbounded"/>
2348     <xs:element ref="muws2:Description"
2349         minOccurs="0" maxOccurs="unbounded"/>
2350     <xs:element ref="muws2:Version"
2351         minOccurs="0"/>
2352
2353     **      Operational Status          **
2354     <xs:element ref="muws2:OperationalStatus"/>
2355
2356     **      Metrics                     **
2357     <xs:element ref="muws2:CurrentTime"/>
2358
2359     **      Relationship                 **
2360     <xs:element ref="muws2:Relationship"
2361         minOccurs="0" maxOccurs="unbounded"/>
2362
2363     **      Relationship Resource        **
2364     <xs:element ref="muws2:Name" minOccurs="0"/>
2365     <xs:element ref="muws2:Type"/>
2366     <xs:element ref="muws2:Participant"
2367         minOccurs="2" maxOccurs="unbounded"/>
2368
2369 -->
2370 </xs:schema>

```

2371

Appendix D. WSDL elements

2372

```
<?xml version="1.0" encoding="utf-8"?>
```

2373

```
<definitions
```

2374

```
  targetNamespace="http://docs.oasis-open.org/wsdm/muws2-2.wsdl"
```

2375

```
  xmlns:muws-p2-wsdl="http://docs.oasis-open.org/wsdm/muws1-2.wsdl"
```

2376

```
  xmlns:muws2="http://docs.oasis-open.org/wsdm/muws2-2.xsd"
```

2377

```
  xmlns:muws1="http://docs.oasis-open.org/wsdm/muws1-2.xsd"
```

2378

```
  xmlns:wsrp="http://docs.oasis-open.org/wsrp-1.xsd"
```

2379

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

2380

```
  xmlns="http://schemas.xmlsoap.org/wsdl/">
```

2381

```
  <types>
```

2382

```
    <xs:schema elementFormDefault="qualified"
```

2383

```
      targetNamespace="http://docs.oasis-open.org/wsdm/muws2-2.wsdl">
```

2384

```
      <xs:import namespace="http://docs.oasis-open.org/wsdm/muws2-2.xsd"
```

2385

```
        schemaLocation="http://docs.oasis-open.org/wsdm/muws2-2.xsd"/>
```

2386

2387

2388

```
      <xs:import namespace="http://docs.oasis-open.org/wsdm/muws1-2.xsd"
```

2389

```
        schemaLocation="http://docs.oasis-open.org/wsdm/muws1-2.xsd"/>
```

2390

2391

```
    </xs:schema>
```

2392

```
  </types>
```

2393

2394

2395

```
  <message name="QueryRelationshipsByTypeRequest">
```

2396

```
    <part name="body" element="muws2:QueryRelationshipsByType"/>
```

2397

2398

```
  </message>
```

2399

```
  <message name="QueryRelationshipsByTypeResponse">
```

2400

```
    <part name="body" element="muws2:QueryRelationshipsByTypeResponse"/>
```

2401

2402

```
  </message>
```

2403

```
</definitions>
```

2404

2405

```
<!--
```

2406

WSDL COPY Material

2407

Copy and paste the operation specification below into a portType definition of

2408

the WSDL documents of a web service.

2409

2410

NOTE: You must import the MUWS WSDL (wsdmmuws2-2).

2411

2412

```
  <operation name="QueryRelationshipsByType">
```

2413

```
    <input name="QueryRelationshipsByTypeRequest"
```

2414

```
      message="muws-p2-wsdl:QueryRelationshipsByTypeRequest"/>
```

2415

```
    <output name="QueryRelationshipsByTypeResponse"
```

2416

```
      message="muws-p2-wsdl:QueryRelationshipsByTypeResponse"/>
```

2417

2418

```
  </operation>
```

2419

2420

```
-->
```

2421

Appendix E. Topics

```
2422 <wstop:TopicSpace name="MuwsNotificationTopics"
2423     targetNamespace="http://docs.oasis-open.org/wsdm/muws2-2.xml"
2424     xmlns:muws1="http://docs.oasis-open.org/wsdm/muws1-2.xsd"
2425     xmlns:muws2="http://docs.oasis-open.org/wsdm/muws2-2.xsd"
2426     xmlns:wstop="http://docs.oasis-open.org/wsn/t-1.xsd"
2427     xmlns:wsrfrp="http://docs.oasis-open.org/wsrfrp-1.xsd">
2428
2429     <wstop:Topic name="IdentityCapability"
2430         messageTypes="muws1:ManagementEvent">
2431     </wstop:Topic>
2432
2433     <wstop:Topic name="ManageabilityCharacteristicsCapability"
2434         messageTypes="muws1:ManagementEvent">
2435     </wstop:Topic>
2436
2437     <wstop:Topic name="CorrelatablePropertiesCapability"
2438         messageTypes="muws1:ManagementEvent">
2439     </wstop:Topic>
2440
2441     <wstop:Topic name="DescriptionCapability"
2442         messageTypes="muws1:ManagementEvent">
2443     </wstop:Topic>
2444
2445     <wstop:Topic name="StateCapability"
2446         messageTypes="muws1:ManagementEvent">
2447     </wstop:Topic>
2448
2449     <wstop:Topic name="OperationalStatusCapability"
2450         messageTypes="muws1:ManagementEvent">
2451     </wstop:Topic>
2452
2453     <wstop:Topic name="MetricsCapability"
2454         messageTypes="muws1:ManagementEvent">
2455     </wstop:Topic>
2456
2457     <wstop:Topic name="ConfigurationCapability"
2458         messageTypes="muws1:ManagementEvent">
2459     </wstop:Topic>
2460
2461     <wstop:Topic name="RelationshipsCapability"
2462         messageTypes="muws1:ManagementEvent">
2463
2464         <wstop:Topic name="RelationshipCreated"
2465             messageTypes="muws1:ManagementEvent">
2466             <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-
2467 19991116">
2468                 //muws1:ManagementEvent[count(muws2:RelationshipCreatedNotification)=1]
2469             </wstop:MessagePattern>
2470         </wstop:Topic>
2471
2472         <wstop:Topic name="RelationshipDeleted"
2473             messageTypes="muws1:ManagementEvent">
2474             <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-
2475 19991116">
2476                 //muws1:ManagementEvent[count(muws2:RelationshipDeletedNotification)=1]
2477             </wstop:MessagePattern>
2478         </wstop:Topic>
2479     </wstop:Topic>
2480 </wstop:TopicSpace>
```

```

2481
2482     <wstop:Topic name="RelationshipResourceCapability"
2483               messageTypes="muws1:ManagementEvent">
2484     </wstop:Topic>
2485
2486     <wstop:Topic name="ManageabilityEndpointCreation"
2487               messageTypes="muws1:ManagementEvent">
2488       <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-
2489 19991116">
2490         //muws1:ManagementEvent[count(muws2:CreationNotification)=1]
2491       </wstop:MessagePattern>
2492
2493       <wstop:Topic name="ManageableResourceCreation"
2494               messageTypes="muws1:ManagementEvent">
2495         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-
2496 19991116">
2497           //muws1:ManagementEvent[count(muws2:CreationNotification)=1]
2498         </wstop:MessagePattern>
2499       </wstop:Topic>
2500
2501     </wstop:Topic>
2502
2503     <wstop:Topic name="ManageabilityEndpointDestruction"
2504               messageTypes="muws1:ManagementEvent">
2505       <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-
2506 19991116">
2507         //muws1:ManagementEvent[count(muws2:DestructionNotification)=1]
2508       </wstop:MessagePattern>
2509
2510       <wstop:Topic name="ManageableResourceDestruction"
2511               messageTypes="muws1:ManagementEvent">
2512         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-
2513 19991116">
2514           //muws1:ManagementEvent[count(muws2:DestructionNotification)=1]
2515         </wstop:MessagePattern>
2516       </wstop:Topic>
2517
2518     </wstop:Topic>
2519
2520 </wstop:TopicSpace>

```

2521 Appendix F. Description of situation types

2522 This appendix defines in more details the situation types introduced in section 2.5.1.

2523 AvailabilitySituation

2524 This category deals with the situations reported from the component, regarding its operational
2525 state and availability. This situation provides a context for operations that can be performed by
2526 the component to establish if a product is installed, operational and ready to process functional
2527 requests, or operational and ready or not ready to process management requests. Existing
2528 message include words like “now ready to take requests”, “online”, and “offline”, for example::

- 2529 • “SOAP connector available at port 8888”

2530

2531 CapabilitySituation

2532 This category is specified when a change in capability of a resource occurs. For example, a
2533 printer has an envelope tray attached to it so that the printer is now has additional paper choices.
2534 The same category would be used if the envelope tray is removed from the printer.

2535

2536 ConfigurationSituation

2537 This category deals with the components identifying configuration changes. Any changes that a
2538 component makes to its configuration should be logged using this category. Existing message
2539 include words like “port number is”, “address is”, and “process id”, for example:

- 2540 • “File transfer configured with host='9.27.11.13', port='9090', securityEnabled='false'”

2541

2542 StopSituation

2543 This category deals with the shutdown process for a component. Messages that indicate that a
2544 component has begun to stop, that it has stopped, or that the stopping process has failed all fall
2545 into this category. Existing messages include words like “stop”, “stopping”, “stopped”,
2546 “completed”, and “exiting”, for example:

- 2547 • “Application stopped: myApp.exe”
- 2548 • “An error occurred while stopping myApp.exe”
- 2549 • “Stopping the JMS provider”

2550

2551 StartSituation

2552 This category deals with the startup process for a component. Messages that indicate that a
2553 component has begun the startup process, that it has finished the startup process, or that it has
2554 aborted the startup process all fall into this category. Existing messages include words like
2555 “starting”, “started”, “initializing”, and “initialized”, for example:

- 2556 • “XYZ protocol support was successfully started”
- 2557 • “XYZ protocol support failed to start”
- 2558 • “Starting EJB: myEjb.jar”

2559

2560 RequestSituation

2561 This category is used in situations that a component uses to identify the completion status of a
2562 request. Typically, these requests are complex management tasks or transactions that a
2563 component undertakes on behalf of a requestor and not the mainline simple requests or
2564 transactions. Existing messages are of the form “*request* started” or “*request* completed” as in
2565 phrases like “configuration synchronization started”, and “backup procedure complete”, for
2566 example:

2567 • “Configuration synchronization completed”
2568 Note that events generated from requests that start up or stop a resource would be categorized
2569 as **StartSituation** or **StopSituation** respectively because they are higher precedent than
2570 **RequestSituation**.

2571

2572 **DestroySituation**

2573 This category deals with the situations occurring when an entity or component was removed or
2574 destroyed. Messages telling that a document was destroyed or a file was deleted all fall into this
2575 category. Existing messages include phrases like “was destroyed”, “about to remove”, and “no
2576 longer exists”, for example:

2577 • “The connection pool was destroyed for data source foo”

2578

2579 **CreateSituation**

2580 This category deals with the situations occurring when a component creates an entity. Messages
2581 telling that a document was created, or a file was created, or an Enterprise JavaBean (EJB) was
2582 created all fall into this category. Existing message include words like was created, about to
2583 create, and now exists, for example:

2584 • “New log file was created”

2585

2586 **DependencySituation**

2587 This category deals with the situations where components cannot find some component or
2588 feature that they require. This category includes messages about not finding the “version” of the
2589 component that was expected. Messages that say a resource was not found, or that an
2590 application or subsystem that was unavailable, also fall into this category. Existing messages
2591 include words like “could not find”, and “no such component”, for example:

2592 • “Error encountered while deploying database schema: no database found”

2593

2594

2595 **ConnectSituation**

2596 This category deals with the situations related to aspects about a connection attempt from one
2597 component to another component. Messages that say a connection failed, that a connection was
2598 created, or that a connection was ended all fall into this category. Existing messages include
2599 words like “connection reset”, “connection failed”, and “failed to get a connection”, for example:

2600 • “Connection creation failed”

2601 • “Connection with http://foo.com created”

2602 • “Failed to close a connection”

2603

2604 **ReportSituation**

2605 This category deals with situations that occur as a result of some setting or occurrence that
2606 causes the resource to asynchronously report various types of data. Types of information that
2607 falls into this category are:

2608

2609 • **Exception related** – some exception has occurred within the resource and it not covered
2610 by any other category.

2611 • **Performance related** – some event occurs, that does not fall into any other category,
2612 that has affected performance in some way. For example, weather conditions may be
2613 affected line quality and network speeds are affected.

- 2614 • **Security related** – some security issue has been detected, like the cabinet door to a
- 2615 secure piece of equipment has been opened or an attack of some sort has been
- 2616 detected.
- 2617 • **Heartbeat related** – the resource has been configured to periodically report a ‘heartbeat’.
- 2618 • **Status related** – some change of status that does not affect availability or capability of
- 2619 the resource has been detected. For example, printer ink cartridge is low.
- 2620 • **Log related** – the resource has been configured to generate a log entry based on some
- 2621 event or at a fixed interval. This category identifies this event as a requested log entry.
- 2622 • **Debug related** – the resource has been enabled to turn on diagnostic information flow
- 2623 and will report the information within this category.
- 2624 • **Trace related** – the resource has been enabled to run trace information and reports this
- 2625 information using this category

2626

2627 **OtherSituation**

2628

2629 This category is for those events that do not fall into any other category. Note that this category
2630 is defined for syntactic completeness but any events placed in this category will not be able to be
2631 effectively correlated and its use is therefore discouraged unless absolutely necessary.

2632

2633