



1

2

3

4

Web Services Distributed Management: Management of Web Services (WSDM-MOWS) 1.0

5

Committee Draft, 10 December 2004

6

Document identifier:

7

cd-wsdm-mows-1.0

8

Location:

9

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

10

Editors:

11

Igor Sedukhin, Computer Associates <igor.sedukhin@ca.com>

12

Abstract:

13

The Web Services Distributed Management (WSDM) specifications, as declared in the committee charter, define A) how management of any resource can be accessed via Web services protocols – Management Using Web Services, or MUWS, and B) management of the Web services resources via the former – Management Of Web Services, or MOWS. This document is the WSDM specification defining MOWS.

14

15

16

17

18

Status:

19

This is a draft document and there is no guarantee any part of its content will appear in the final release specification. This document is updated periodically on no particular schedule. Send editorial comments to the editor.

20

21

22

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

23

24

25

26

27

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

28

29

30

31

Since this specification is not yet final, there are no errata available.

32 Table of Contents

33	1	Introduction	3
34	1.1	Terminology.....	3
35	1.2	Notational conventions.....	4
36	2	Architecture.....	5
37	2.1	In-band and Out-of-band Manageability	6
38	2.2	Application to Resources Exposed as Web Services	6
39	2.3	Self-Management.....	7
40	2.4	Formal Representation of the Architecture	8
41	3	Managing Web Services	9
42	3.1	Responsibilities of the Implementations of the Manageability Endpoints	9
43	3.2	Manageability at the Web service level.....	10
44	3.3	Using manageability of Web services endpoints	10
45	4	Security Considerations	12
46	4.1	Additional security considerations when managing Web services	12
47	5	Web service manageability capabilities	14
48	5.1	Common manageability capabilities.....	15
49	5.1.1	Manageability References	15
50	5.2	Web service endpoint manageability capabilities	16
51	5.2.1	Identity	16
52	5.2.2	Identification.....	17
53	5.2.3	Metrics	18
54	5.2.4	Operational State	21
55	5.2.5	Operational Status	24
56	5.2.6	Request Processing State	24
57	6	References.....	35
58	6.1	Normative	35
59	6.2	Non-normative.....	35
60		Appendix A. Acknowledgments	37
61		Appendix B. Revision History	38
62		Appendix C. Notices	39
63		Appendix D. XML Schemas.....	40
64		Appendix E. WSDL elements	47
65		Appendix F. Notification topic spaces	48

66

67 1 Introduction

68 Web services are an integral part of the IT landscape, and, as such, are vital resources to many
69 organizations. Web services may interact with other Web services and are used in business
70 processes. Interacting Web services form a logical network which may span enterprise
71 boundaries. Managing such a logical network is critical for organizations that use Web services to
72 automate and integrate various internal functions, and deal with partners and clients
73 electronically. To manage the Web services network, one needs to manage the components that
74 form the network – the Web services endpoints. This part of the WSDM specification addresses
75 management of the Web services endpoints using Web services protocols **[MOWS-Reqs]**.

76

77 The *Management Of Web Services* (MOWS) specification is based on the concepts and
78 definitions expressed in the *Management Using Web Services* specification (MUWS) **[MUWS]**. It
79 is recommended that the reader is aware of the MUWS specification contents.

80

81 Definitions and examples in this document are based on the following specifications. It is
82 recommended that the reader is aware of their contents.

- 83 ▪ WS Architecture **[WS-Arch]**
- 84 ▪ XML **[XML]**
- 85 ▪ XML Namespaces **[XNS]**
- 86 ▪ XML Schema **[XMLS]**
- 87 ▪ SOAP **[SOAP]**
- 88 ▪ WSDL **[WSDL]**
- 89 ▪ WS-Addressing **[WS-A]**
- 90 ▪ WS-ResourceProperties **[WS-RP]**
- 91 ▪ WS-BaseNotification **[WS-N]**
- 92 ▪ WS-Topics **[WS-T]**
- 93 ▪ XML Path Language **[XPath]**

94

95 Section 5 and appendices D, E and F are *normative* specifications. The rest of the document is
96 *non-normative*, and is provided as a background and explanatory material.

97

98 1.1 Terminology

99 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
100 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be
101 interpreted as described in **[RFC2119]**.

102

103 This specification is based on the terminology defined in the WSDM **[MUWS]** specifications. In
104 addition, the following terms are defined.

105 ***Manageable Web service endpoint*** – is a Web service endpoint as a manageable resource.

106

107 1.2 Notational conventions

108 This specification uses an informal syntax to describe the XML grammar of the messages,
109 property instances and event information forming the manageability capability interfaces. This
110 syntax uses the following rules:

- 111 ▪ The syntax appears as an XML instance, but the values indicate the data types instead of
112 values.
- 113 ▪ {any} is a placeholder for elements from some other namespace (like ##other in XML
114 Schema).
- 115 ▪ Characters are appended to attributes, elements, and {any} to indicate the number of
116 times they may occur as follows: ? (0 or 1), * (0 or more), + (1 or more). No character
117 indicates exactly 1 occurrence. The characters [and] are used to indicate that contained
118 items are to be treated as a group with respect to the ?, *, and + characters.
- 119 ▪ Attributes, elements, and values separated by | and grouped with (and) are meant to be
120 syntactic alternatives.
- 121 ▪ ... is used in XML start elements to indicate that attributes from some other namespace
122 are allowed.
- 123 ▪ The XML namespace prefixes are used to indicate the namespace of the element being
124 defined

125 A full WSDL description of all interfaces and XML Schemas of all information elements are
126 available in the appendices.

127

128 When describing instances of XML information, and in order to refer to elements and attributes,
129 this specification uses a simplified XPath [**XPath**] notation which can be formally defined as
130 follows.

- 131 ▪ Path = '/'? (['@' ? (NCName | QName | '*')] | ['(' (NCName | QName | '*') ')'] ['/' Path] ?
- 132 ▪ NCName is an XML non-qualified name as defined by XML Schema [XMLS]. In this case
133 the namespace is assumed to default to the namespace of this specification.
- 134 ▪ QName is an XML qualified name as defined by XML Schema [XMLS].
- 135 ▪ The symbol * denotes any name match.
- 136 ▪ The symbol / denotes a path delimiter. If it appears as the first element of the path, it
137 denotes the root of the XML document.
- 138 ▪ The symbol @ denotes a reference to an XML attribute, otherwise NCName, QName or *
139 refer to an XML element.
- 140 ▪ The symbols (and) denote a reference to an XML Schema type.

141

142 For example, /E1/E2/@A1 refers to an attribute A1 of an element E2 contained in element E1
143 which is a root of the XML document. E1/ns1:E2/E3 refers to an element E3 which is contained in
144 the element E2 which is contained in the element E1 anywhere in the XML document. In this case
145 element E2 belongs to the namespace mapped to the prefix ns1. (ns2:T1)/E1/ns1:E2/@A1 refers
146 to an attribute A1 on an element E2 contained in the element E1 declared in the XML Schema
147 type T1 which target namespace is mapped to the prefix ns2.

148

149

2 Architecture

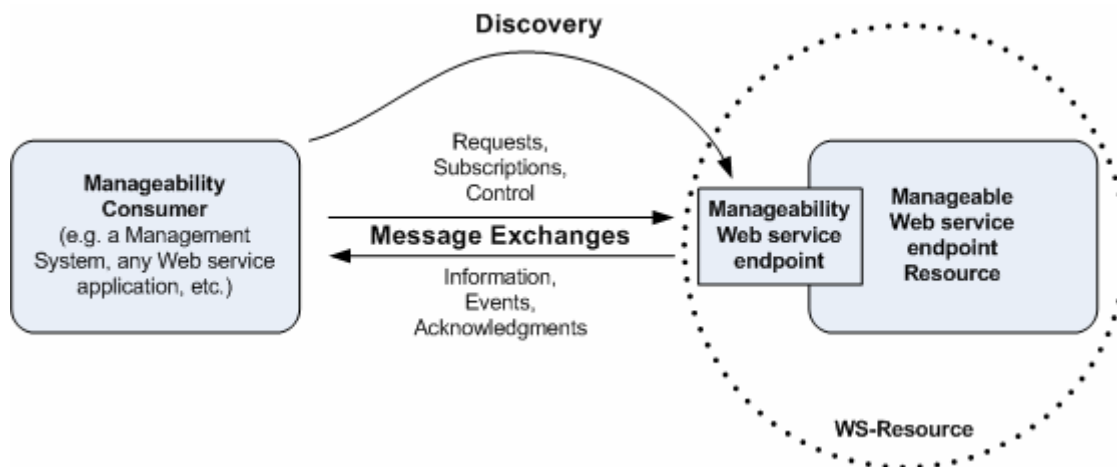
150

151 Management of Web services (MOWS) is an application of Management using Web services
152 (MUWS) to the resources which are elements of the Web Services Architecture [WS-Arch]. This
153 WSDM specification defines how the manageability of Web service endpoints and resources
154 exposed as Web services can be accessed via Web services. In order to achieve this goal,
155 MOWS is based on the MUWS specifications, and the architecture, definitions and dependencies
156 thereof [MUWS].

157

158 Application of the WSDM architecture concepts (§2 of the MUWS specification part 1) to the
159 management of Web services could be described as follows (Figure 1). A *manageability Web*
160 *service endpoint* (or, shortly, *manageability endpoint*) provides access to the *manageable Web*
161 *service endpoint resource* (a manageable resource, in terms of MUWS). A manageable Web
162 service endpoint (or, shortly, *manageable endpoint*) could be, for example, an endpoint of an
163 order entry Web service for which received messages could be counted and reported to the
164 *manageability consumers*. Following the WSDM concepts, the manageability consumer discovers
165 the manageability endpoint and exchanges messages with it in order to request information,
166 subscribe to events or control the manageable endpoint resource.

167



168

169

Figure 1. Management of Web services concepts

170

171 Refer to the §2 of the MUWS specification part 1 [MUWS] for more detailed explanation of
172 discovery and message exchange between manageability consumers and manageability
173 endpoints.

174

175 Understanding of a number of other important aspects of the WSDM architecture may be
176 necessary. Please refer to the following sections of the MUWS specification [MUWS]

177

- 178 ▪ **Focus on resources** (§2.1 of MUWS part 1) – focus on providing access to the
179 manageable resources – a contract between a manageability consumer and a
manageable resource with regards to discovery and message exchanges.
- 180 ▪ **Composeability** (§2.2 of MUWS part 1) – allows a non-conflicting, incremental mix of
181 Web services implementation aspects and manageability capabilities.

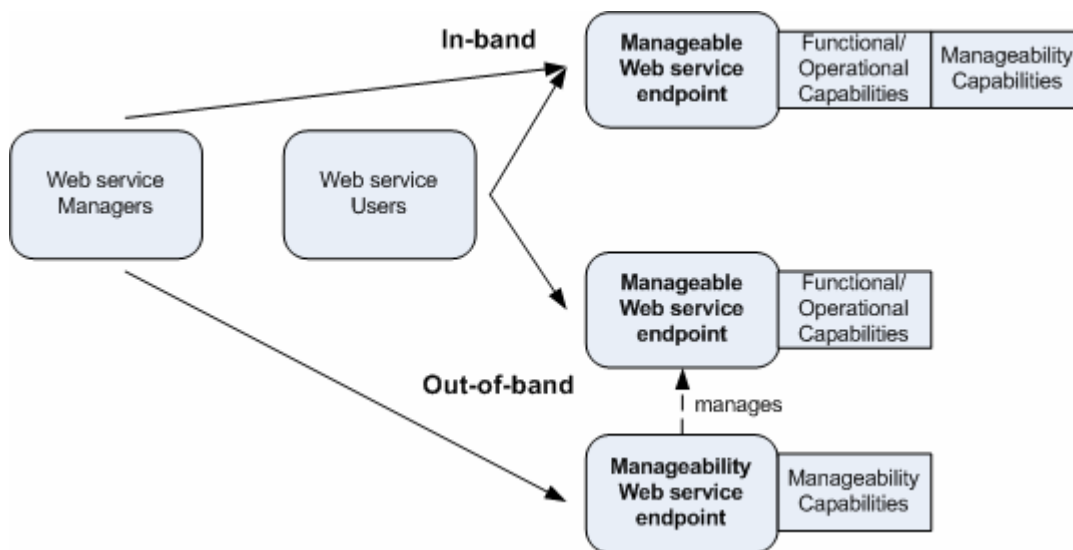
182

2.1 In-band and Out-of-band Manageability

183

An interesting peculiarity of the MOWS subject domain is that a manageability endpoint and a manageable endpoint are both Web services endpoints, and therefore could be the same endpoint or could be different endpoints. In other words, manageability consumers and regular Web service consumers could target their messages to the same or to different endpoints. Either of the approaches is allowed by the MOWS architecture and the implementation choices are transparent for manageability consumers (and Web service consumers, for that matter). The Figure 2 illustrates this.

189



191

192

193

Figure 2. In-band and out-of-band manageability

194

2.2 Application to Resources Exposed as Web Services

195

WSDM allows a resource and all of its services to be manageable in a standard and interoperable manner. A resource may support both manageability and functional capabilities. For example, a printer can obviously print, but the same printer may also be able to indicate if it is on-line and may be able to notify when the toner is running out. A manageable resource may allow access to its manageability capabilities and functional capabilities via Web services. Web services represent a composition of manageable and functional qualities of a given resource (Figure 3).

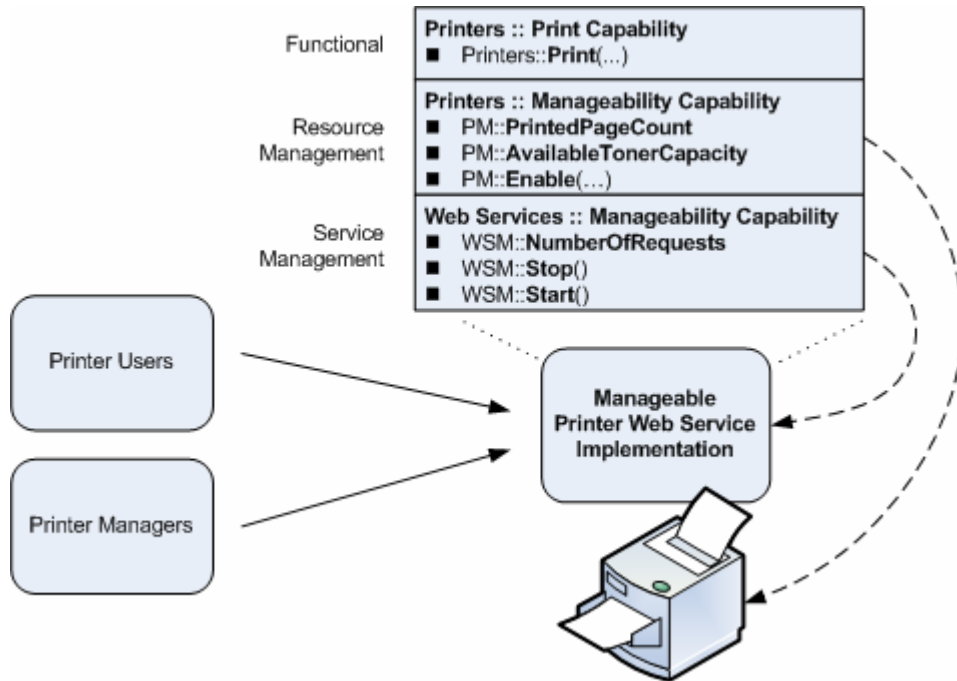
201

Manageability consumers might take advantage of a composition of manageability and functional capabilities: 1) management-oriented consumers gain visibility into functional aspects of a resource 2) business-oriented consumers gain visibility into management aspects of a resource. For example, a Web services-based business process may involve a selection of an on-line printer with good amount of toner in order to print an urgent report for executives.

206

Composeability makes it easy for implementers of resource services to offer an appropriate set of functional capabilities along with an appropriate set of manageability capabilities guided by the appropriate model for authorization of these requests.

209

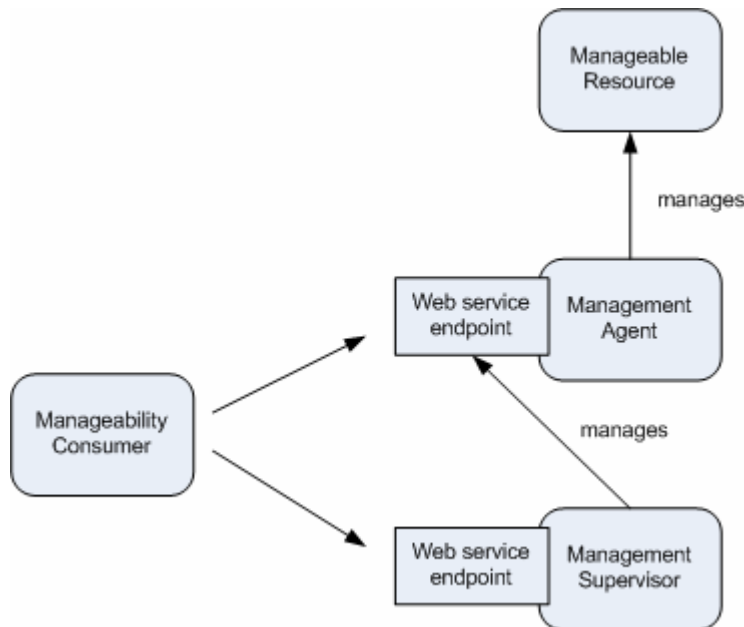


210
211

Figure 3.Application to resources exposed as Web services

212 **2.3 Self-Management**

213 The WSDM specifications define how to use Web services to expose manageable resources
 214 (MUWS), and in addition, define how to expose manageable Web service implementations
 215 (MOWS – this specification). Application of MOWS to MUWS gives an interesting combination of
 216 the manageable management. Using both specifications, it is possible to build reliable and
 217 accountable management systems (Figure 4).



218
219
220

Figure 4.Applying MOWS to MUWS

228 3 Managing Web Services

229 Using definitions expressed in WSDL 1.1 [WSDL] and WS-Addressing [WS-A] as guidelines, a
230 Web service (described by a WSDL 1.1 service element) is an aggregate of endpoints (described
231 by WSDL 1.1 port elements). An endpoint binds a Web service interface (described by a WSDL
232 1.1 portType element) to an address (URI). Each interface describes a set of messages that
233 could be exchanged and their format. Properly formatted messages could be sent to the endpoint
234 at the address in the way prescribed by the binding (described by a WSDL 1.1 binding element).
235 A Web service description contains definitions of a combination of interfaces and services.

236

237 According to the §2, management of Web services starts at an endpoint resource which,
238 therefore, becomes a manageable resource, specifically called a manageable endpoint. The
239 reason the Web service endpoint is the basic manageable resource is that (1) anything behind an
240 endpoint is a concrete implementation (e.g. an application hosted on a server), and (2) an
241 aggregate of endpoints is a logical construct, management of which has to be inferred from
242 manageability of the constituent endpoints. This specification focuses on defining manageability
243 capabilities of the Web service endpoints. Furthermore, (1) is in the realm of the
244 applications/systems/networks management, and (2) should be done by the intelligent
245 management systems. Aspects of (1) are further discussed in §3.1. Aspects of (2) are further
246 discussed in §3.2.

247

248 This specification balances requirements of Web services management applications and the
249 complexity of implementing manageability endpoints.

250 3.1 Responsibilities of the Implementations of the Manageability 251 Endpoints

252 The system providing manageability capabilities for a Web service endpoint must be aware of the
253 environment as experienced from the Web service caller's point of view. This *experience* may be
254 dependent upon hardware or software configuration in which the Web service endpoint exists.
255 Implementations of manageability endpoints may need to account for management requests
256 made with respect to the Web service caller's point of view.

257

258 Consider two examples. The first case is that of a hardware routing configuration. A hardware
259 device controls access to all messages sent to a particular URL such as
260 `http://external.example.com/theService`. Upon receipt of messages for that URL, the device
261 distributes the messages to Web service endpoints at the `http://s1.example.com/theService`,
262 `http://s1.example.com/theService`, and `http://s2.example.com/theService` addresses.

263

264 If, say, a query regarding metrics were made regarding the Web service endpoint receiving
265 messages at the `http://external.example.com/theService` address, it is the responsibility of the
266 implementation of the manageability endpoint to aggregate the results from the three underlying
267 Web service endpoints to provide a meaningful response.

268

269 A second example is one wherein a single Web service endpoint is accessible at two distinct
270 URLs due to DNS aliasing. Consider the Web service endpoint at
271 `http://services.example.com/creditCheck`. External to the Example Company, this Web service
272 endpoint is accessible at the `http://ourservices.example.com/creditCheck` address, while
273 internally, this Web service endpoint is accessible at
274 `http://extservices.example.com/creditCheck`. However, in both cases, the message processing is

275 performed by the same machine, application, code, etc. The Web service endpoint
276 implementation itself is *aware* of the means by which it is addressed (e.g. is using the URL
277 header of the HTTP messages), and it adjusts message processing appropriately.

278

279 In this case, the implementation of the manageability endpoint must be similarly aware of how the
280 Web service endpoint was accessed. Queries regarding the two URL aliases must be accounted
281 for separately, even though the underlying Web service endpoint is the same.

282 **3.2 Manageability at the Web service level**

283 Management applications may want to manage Web services at the granularity level of the
284 endpoint. For example, to find out when an endpoint goes down and how many messages a
285 specific endpoint has processed. At the same time, there are many cases where the
286 management applications may want to manage the Web service as a logical aggregate of all of
287 its endpoints. For example, a business manager using a business dashboard doesn't care
288 whether the purchase orders arrive via the HTTP or the SMTP binding of the order entry Web
289 service, or whether orders arrive via the US server or its European mirror.

290

291 In recognition of these requirements, this specification defines manageability of endpoints as the
292 base building block for managing Web services. The specification ensures that information is
293 available to management applications in order to summarize to the Web service-level view. This
294 includes allowing manageable endpoints to establish relationships linking them as part of the
295 same Web service.

296 **3.3 Using manageability of Web services endpoints**

297 The following pattern may be used by the manageability consumers which intend to manage Web
298 services endpoints.

- 299 1. Obtain an EPR to the manageability endpoint. One of the following ways may be used.
 - 300 a. Discover manageable resources as described in the MUWS specifications
301 **[MUWS]**.
 - 302 b. Exercise the Manageability References capability (§5.1.1) on the functional Web
303 services endpoint.
 - 304 c. The functional Web services endpoint may also be the manageability endpoint
305 (§2.1). Determine that by detecting if the endpoint supports the MUWS Identity
306 capability:
 - 307 i. Either, obtain the WSDL document describing the manageability
308 endpoint and look for a ResourceId element (see MUWS specification
309 part 1 §5.1) in the first level children of the resource properties document
310 root **[WS-RP]**.
 - 311 ii. Or, request the value of the ManageabilityCapability property (see
312 MUWS specification part 1 §5.2) and look for the URI which identifies the
313 MUWS Identity capability.
- 314 2. Using the EPR obtained in the previous step, and based on the manageability capabilities
315 intended to be used, build Web services messages targeted at the manageable Web
316 services endpoint.
 - 317 a. Obtain the WSDL document describing the manageability endpoint and
318 understand how operations defined by the manageability capabilities are bound.
 - 319 b. Request the value of the ManageabilityCapability property (see MUWS
320 specification part 1 §5.2) and look for the URIs which identify the capabilities to
321 be used.

322
323
324
325

- c. To understand how to construct Web services messages for management of a Web services endpoint, consult the manageability capability definition sections in this specification or in the MUWS specification and any dependent specifications thereof.

326 4 Security Considerations

327 It is RECOMMENDED that communication between a manageability consumer and a
328 manageability endpoint be secured using the mechanisms described in WS-Security [WSS] and
329 WS-I Basic Security Profile [BSP], including transport-level security such as HTTP over Secure
330 Socket Layers (SSL). In order to properly secure messages, the body and all relevant headers
331 may need to be signed and encrypted.

332 The following list summarizes common classes of attacks that apply generally to protocols and
333 identifies mechanisms available to prevent/mitigate the attacks:

- 334 ▪ **Message alteration** – Alteration is prevented by including signatures of the message
335 information using WS-Security.
- 336 ▪ **Message disclosure** – Confidentiality is preserved by encrypting sensitive data using
337 WS-Security.
- 338 ▪ **Key integrity** – Key integrity is maintained by using the strongest algorithms possible.
- 339 ▪ **Authentication** – Authentication is established using the mechanisms described in WS-
340 Security and WS-Trust. Each message is authenticated using the mechanisms
341 described in WS-Security.
- 342 ▪ **Accountability** – Accountability is a function of the type of and strength of the key and
343 algorithms being used. In many cases, a strong symmetric key provides sufficient
344 accountability. However, in some environments, strong PKI signatures are required.
- 345 ▪ **Availability** – All services are subject to a variety of availability attacks. Replay detection
346 is a common attack and it is RECOMMENDED that this be addressed by the
347 mechanisms described in WS-Security. Other attacks, such as network-level denial of
348 service attacks are harder to avoid and are outside the scope of this specification. That
349 said, care should be taken to ensure that minimal state is saved prior to any
350 authenticating sequences.

351

352 The WS-I Basic Security Profile working group has produced a scenarios document which
353 explores these threats in more detail and which identifies security requirements which are then
354 addressed by subsequent profiles [BSP]. WSDM looks to the security domain experts to define
355 the mechanisms to secure web services and looks to WS-I to define interoperability profiles that
356 can be leveraged by WSDM implementers.

357

358 4.1 Additional security considerations when managing Web 359 services

360 It is RECOMMENDED that the implementers of manageability endpoints and manageability
361 consumers take into consideration the following security related concerns.

- 362 ▪ If a manageable Web services endpoint supports messages from both a consumer of a
363 service and a manager of a service §2.1, it may be important to identify a security model
364 which allows for the appropriate level of granularity with regard to the message origin. For
365 example, setting configuration options may be allowed by a manageability consumer but
366 not an application consumer. When these composed services are deployed, it will be
367 important to understand the authorization model for both management and functional
368 use.
- 369 ▪ In order to make the management systems secure in addition to reliable and accountable
370 (§2.3), it will be important to follow a set of guidelines and best practices that detail how

371 to compose MOWS with existing security implementations and emerging specifications
372 for authorization and trust.

373 ■ Implementers of this specification may need to give a particular attention to security when
374 implementing the following manageability capabilities.

375 ○ Manageability References (§5.1.1) – this capability allows access to the
376 manageability endpoint references of a functional Web service endpoint. The
377 concern is that visibility to these references may need to be protected differently
378 than visibility of the functional Web service endpoint and its operations.

379 ○ Request Processing State (§5.2.6) – this capability allows managers to subscribe
380 to notifications against request processing by a functional Web service endpoint.

381 1. Not all managers should be allowed to subscribe to request processing
382 notification because messages may contain protected information, and/or
383 may be used to generate a DoS attack.

384 2. The request messages may be encrypted and signed. Therefore, managers
385 may need to possess information that allows them to deal with such
386 encrypted and signed messages.

387 3. Notification messages which contain information about request messages
388 SHOULD be encrypted to avoid spoofing of this information by intercepting
389 notification messages.

390 4. The request processing notification message provides sufficient flexibility
391 with respect to its content to avoid inclusion of information which needs to be
392 highly protected and therefore not relayed to managers.

393

5 Web service manageability capabilities

394

395 The following sections define manageability capabilities for Web services and resources exposed
396 as Web services (see 2.2).

397

398 Each capability is described in a UML summary diagram. Metadata is defined for properties,
399 operations and events according to MUWS specification part 1 §3.4 and part 2 §2.4 **[MUWS]**.

400

401 The definitions of the Web service manageability capabilities are rendered into WSDL elements
402 (interfaces/portTypes) and supporting XML Schemas in Appendix D and Appendix E. Appendix F
403 contains renditions of the notification topic spaces for the events defined by the capability
404 specifications.

405

406 Following namespace prefixes are used in this document when referring to XML elements and
407 XML schemas. The table below describes what prefix corresponds to which namespace URI.

408

Prefix	Namespace
muws-xs1	http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part1.xsd
muws-xs2	http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part2.xsd
muws-wsdl	http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part2.wsdl
mows-xs	http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.xsd
mows-wsdl	http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.wsdl
mows-events	http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows-events.xml
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing
wsdl	http://www.w3.org/2002/07/wsdl
S	http://www.w3.org/2002/12/soap-envelope
xs	http://www.w3.org/2001/XMLSchema
wsrf-rp	http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-1.2-draft-01.xsd
wsnt	http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.xsd
wstop	http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.xsd

409

410 Unless otherwise specified, XML elements and XML schema types introduced in this specification
411 belong to the namespace mapped to the **mows-xs** prefix.

412

413 5.1 Common manageability capabilities

414 The following sections define manageability capabilities applicable to Web services and
415 resources exposed as Web services.

416 5.1.1 Manageability References

417 This capability is identified by the following URI:

418 <http://docs.oasis-open.org/wsdm/2004/12/mows/capabilities/ManageabilityReferences>

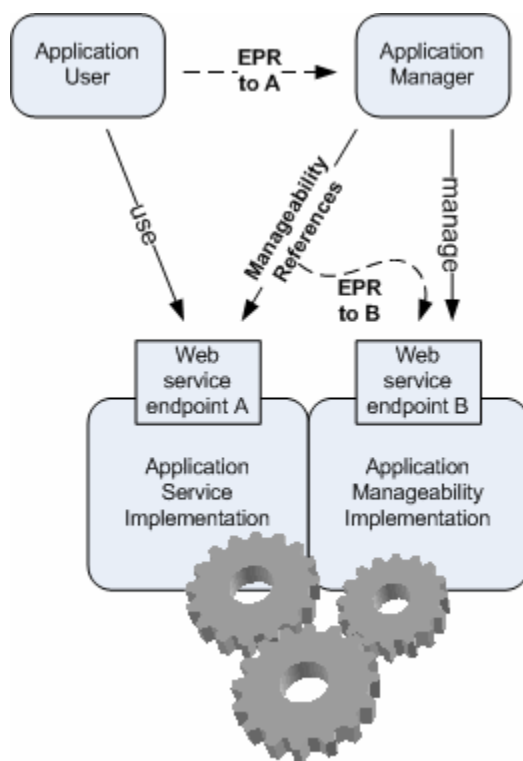
419

420 This capability allows a functional/operational Web service or a resource exposed as a Web
421 service (§2.2) (*the service*) to provide references to its manageability endpoints. This capability is
422 intended for implementations of functional/operational Web services endpoints. The consumer
423 may exchange messages with *the service* in order to request references to the manageability
424 endpoints. Using obtained references, the consumer may exchange messages with the
425 manageability endpoints in order to perform management activities to *the service*.

426

427 For example (Figure 6), an application user accesses a Web service endpoint A. The application
428 user then gives the endpoint A reference to the application manager which accesses the Web
429 service endpoint A in order to obtain a reference to the application manageability implementation
430 accessible at the Web service endpoint B. The application manager may now manage the
431 application by exchanging management related messages with endpoint B.

432



433

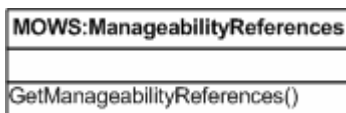
434

Figure 6. Use of Manageability References capability

435

436 The Manageability References capability is represented by the **ManageabilityReferences** UML
437 model class. The name of this class identifies the semantics of this capability.

438



439

440

Figure 7. Manageability References capability model

441

442 5.1.1.1 Operations

443 The following is the specification of the Manageability References capability operations.

444

445 5.1.1.1.1 GetManageabilityReferences

446 This operation is mandatory for implementations of this capability and is defined as the following
447 message exchange.

448

449 The request to perform this operation is a message containing the following XML element.

450

```
451 <GetManageabilityReferences/>
```

452 **GetManageabilityReferences** is a Global Element Declaration (GED) which identifies the
453 request of the GetManageabilityReferences operation.

454

455 The response to the above request is either a fault (any fault) or a message containing the
456 following XML element.

457

```
458 <GetManageabilityReferencesResponse>  
459 <muws-xs1:ManageabilityEndpointReference>  
460 <!-- see [MUWS] -->  
461 </muws-xs1:ManageabilityEndpointReference>+  
462 </GetManageabilityReferencesResponse>
```

463

464 **GetManageabilityReferencesResponse** is a GED which identifies the response to the
465 requested GetManageabilityReferences operation.

466

467 **GetManageabilityReferencesResponse/muws-xs1:ManageabilityEndpointReference** is a
468 reference to the Web service endpoint which provides access to the management of the
469 functional/operational Web service endpoint or the Web service-enabled resource which
470 responded to the GetManageabilityReferences operation request message.

471

472 5.2 Web service endpoint manageability capabilities

473 The following sections define manageability capabilities applicable to Web service endpoints.

474 5.2.1 Identity

475 A WSDM manageable endpoint MUST support the MUWS **Identity** manageability capability (§5.1
476 of the [MUWS] part 1). There are no extensions to the MUWS definition of this capability.

477 5.2.2 Identification

478 This capability is identified by the following URI:

479 <http://docs.oasis-open.org/wsdm/2004/12/mows/capabilities/Identification>

480 All properties, operations and events defined for this capability have the following metadata:

- 481 ▪ `<muws-xs2:Capability>http://docs.oasis-`
- 482 `open.org/wsdm/2004/12/mows/capabilities/Identification</muws-xs2:Capability>`

483

484 The Web service endpoint's manageable identification capability is represented by the
485 **Identification** UML model class. The name of the class identifies the semantics of this capability.
486 This capability name and semantics are consistent with the following definition (from the Webster
487 dictionary).

488 identification: **1 a** : an act of identifying : the state of being identified **b** : evidence of
489 identity

490

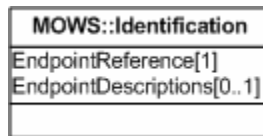
491 Note that, in contrast, the MUWS **Identity** capability and semantics are consistent with the
492 following definition (from the Webster dictionary).

493 identity: **1 a** : sameness of essential or generic character in different instances **b** :
494 sameness in all that constitutes the objective reality of a thing : ONENESS

495

496 The *identification* capability is used to help establish the Web service endpoint being managed.
497 The *identity* capability may be used to determine if two manageability endpoints provide
498 manageability of the same resource or not.

499



500

501 **Figure 8.** Endpoint identification manageability capability model

502

503 5.2.2.1 Properties

504 The following is the specification of the Web service endpoint identification properties (i.e. XML
505 elements which represent properties).

506

507 `<EndpointReference>wsa:EndpointReferenceType</EndpointReference>`

508 `<EndpointDescriptions><description>xs:anyURI</description>*</EndpointDescriptions>?`

509

510 **EndpointReference** is a reference to the Web service endpoint being managed. A reference
511 must be resolvable to the actual useable endpoint. This property represents one way to access
512 the endpoint resource but doesn't preclude the existence of multiple descriptions of the same
513 endpoint resource. Metadata about this property is as follows.

- 514 ▪ Is not *Mutable*
- 515 ▪ Is not *Modifiable*

516 **EndpointDescriptions** is a list of URIs pointing to description documents of the Web service
517 endpoint resource. The different description documents can be of the same or of different types
518 (e.g. WSDL1.1, WSDL2.0, UDDI tModel, etc.). Metadata about this property is as follows.

- 519 ▪ Is *Mutable*
- 520 ▪ Is not *Modifiable*

521 **5.2.2.2 Events**

522 The following specification defines this capability notification topics in the namespace mapped to
 523 the **mows-events** prefix.

524

```
525    <wstop:Topic name="IdentificationCapability" messageTypes="muws-xs1:ManagementEvent"/>
```

526

527 **mows-events:IdentificationCapability** is a topic on which management events related to this
 528 manageability capability SHOULD be emitted.

529

530 **5.2.3 Metrics**

531 This capability is identified by the following URI:

532 <http://docs.oasis-open.org/wsdm/2004/12/mows/capabilities/Metrics>

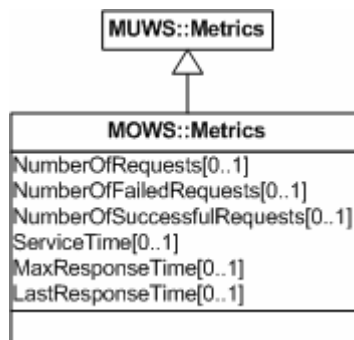
533 All properties, operations and events defined for this capability have the following metadata:

- 534 ▪ <muws-xs2:Capability>[http://docs.oasis-](http://docs.oasis-open.org/wsdm/2004/12/mows/capabilities/Metrics)
 535 open.org/wsdm/2004/12/mows/capabilities/Metrics</muws-xs2:Capability>

536

537 The Web service endpoint's manageable metrics capability is represented by the **Metrics** UML
 538 model class. The name of the class identifies the semantics of this capability.

539



540

541 **Figure 9.** Endpoint metrics manageability capability model

542

543 This capability extends the definition of the MUWS Metrics capability. WSDM manageable
 544 endpoints that intend to support the MOWS **Metrics** capability MUST support the MUWS **Metrics**
 545 capability (§3.4 of the **[MUWS]** part 2) as well.

546

547 It is recommended that for adequate calculations, the Web service endpoint metric properties
 548 (one or all) are retrieved together with the **muws-xs2:CurrentTime** property (e.g., using one
 549 request to retrieve multiple properties).

550

551 Metrics and request processing states are related. The request processing state change
 552 boundaries are the points where metric counters are incremented. These states are defined
 553 below, in §5.2.6.

554 **5.2.3.1 Information markup declarations**

555 The following two XML Schema complex types are defined for metrics that represent integers and
556 durations of time.

557

```
558 <xs:complexType name="IntegerCounter">  
559     <xs:simpleContent>  
560         <xs:extension base="xs:nonNegativeInteger">  
561             <xs:attributeGroup ref="muws-xs2:MetricAttributes"/>  
562             <xs:anyAttribute namespace="##other" processContents="lax"/>  
563         </xs:extension>  
564     </xs:simpleContent>  
565 </xs:complexType>
```

566

567 **(IntegerCounter)** type declares an xs:nonNegativeInteger counter metric.

568

```
569 <xs:complexType name="DurationMetric">  
570     <xs:simpleContent>  
571         <xs:extension base="xs:duration">  
572             <xs:attributeGroup ref="muws-xs2:MetricAttributes"/>  
573             <xs:anyAttribute namespace="##other" processContents="lax"/>  
574         </xs:extension>  
575     </xs:simpleContent>  
576 </xs:complexType>
```

577

578 **(DurationMetric)** type declares an xs:duration metric.

579 **5.2.3.2 Properties**

580 The following is the specification of the Web service endpoint metrics properties (i.e. XML
581 elements which represent properties).

582

```
583 <NumberOfRequests>IntegerCounter</NumberOfRequests>?  
584 <NumberOfFailedRequests>IntegerCounter</NumberOfFailedRequests>?  
585 <NumberOfSuccessfulRequests>IntegerCounter</NumberOfSuccessfulRequests>?  
586 <ServiceTime>DurationMetric</ServiceTime>?  
587 <MaxResponseTime>DurationMetric</MaxResponseTime>?  
588 <LastResponseTime>DurationMetric</LastResponseTime>?
```

589

590 **NumberOfRequests** is a counter of the number of request messages that the Web service
591 endpoint has received. This counter is incremented by 1 whenever a request reaches the
592 Received state according to the Figure 12. Metadata about this property is as follows.

593

- Is *Mutable*

594

- Is not *Modifiable*

595

- <muws-xs2:ChangeType>Counter</muws-xs2:ChangeType>

596

- <muws-xs2:TimeScope>SinceReset</muws-xs2:TimeScope> or <muws-
597 xs2:TimeScope>Interval</muws-xs2:TimeScope>

598 **NumberOfFailedRequests** is a counter of the number of request messages that the Web service
599 endpoint has received, and a (SOAP) fault was sent in reply. This counter is incremented by 1
600 whenever a request reaches the Failed state according to the Figure 12. Metadata about this
601 property is as follows.

- 602 ▪ *Is Mutable*
- 603 ▪ *Is not Modifiable*
- 604 ▪ `<muws-xs2:ChangeType>Counter</muws-xs2:ChangeType>`
- 605 ▪ `<muws-xs2:TimeScope>SinceReset</muws-xs2:TimeScope>` or `<muws-`
- 606 `xs2:TimeScope>Interval</muws-xs2:TimeScope>`

607 **NumberOfSuccessfulRequests** is a counter of the number of request messages that the Web
 608 service endpoint has received, and anything but a (SOAP) fault was sent in reply. This counter is
 609 incremented by 1 whenever a request reaches the Completed state according to the Figure 12.
 610 Metadata about this property is as follows.

- 611 ▪ *Is Mutable*
- 612 ▪ *Is not Modifiable*
- 613 ▪ `<muws-xs2:ChangeType>Counter</muws-xs2:ChangeType>`
- 614 ▪ `<muws-xs2:TimeScope>SinceReset</muws-xs2:TimeScope>` or `<muws-`
- 615 `xs2:TimeScope>Interval</muws-xs2:TimeScope>`

616

617 Note that **NumberOfSuccessfulRequests + NumberOfFailedRequests ≤ NumberOfRequests**
 618 as there could possibly be some requests that were received, but lost or still being processed.

619

620 **ServiceTime** is a counter of the total elapsed time (in seconds) that the Web service endpoint
 621 has taken to process all requests (successfully or not). Metadata about this property is as follows.

- 622 ▪ *Is Mutable*
- 623 ▪ *Is not Modifiable*
- 624 ▪ `<muws-xs2:ChangeType>Counter</muws-xs2:ChangeType>`
- 625 ▪ `<muws-xs2:TimeScope>SinceReset</muws-xs2:TimeScope>` or `<muws-`
- 626 `xs2:TimeScope>Interval</muws-xs2:TimeScope>`

627 **MaxResponseTime** is a gauge indicating the maximum time duration (in seconds) between all
 628 requests received and their completion or failure. Metadata about this property is as follows.

- 629 ▪ *Is Mutable*
- 630 ▪ *Is not Modifiable*
- 631 ▪ `<muws-xs2:ChangeType>Gauge</muws-xs2:ChangeType>`
- 632 ▪ `<muws-xs2:TimeScope>SinceReset</muws-xs2:TimeScope>` or `<muws-`
- 633 `xs2:TimeScope>Interval</muws-xs2:TimeScope>`

634 **LastResponseTime** is a gauge indicating the last recorded time duration (in seconds) between
 635 the last request received and its completion or failure. Metadata about this property is as follows.

- 636 ▪ *Is Mutable*
- 637 ▪ *Is not Modifiable*
- 638 ▪ `<muws-xs2:ChangeType>Gauge</muws-xs2:ChangeType>`
- 639 ▪ `<muws-xs2:TimeScope>PointInTime</muws-xs2:TimeScope>`

640

641 Note that if a metric property has a `<muws-xs2:TimeScope>SinceReset</muws-xs2:TimeScope>`
 642 metadata value, the `muws-xs2:ResetAt` attribute MUST be reported on the property element and
 643 the `muws-xs2:Duration` attribute MUST NOT be reported. If a metric property has a `<muws-`
 644 `xs2:TimeScope>Interval</muws-xs2:TimeScope>` metadata value, the `muws-xs2:ResetAt`
 645 attribute MAY be reported on the property element and the `muws-xs2:Duration` attribute MUST be
 646 reported.

647

648 Also note that in this specification, counters are not just monotonically increasing variables, but
649 also represent a cumulative metric of some kind e.g. number of requests over time. Gauges, on
650 the other hand, do not represent a cumulative metric, and rather represent values of some kind
651 (e.g. response time).

652 5.2.3.3 Events

653 The following specification defines this capability notification topics in the namespace mapped to
654 the **mows-events** prefix.

655

```
656 <wstop:Topic name="MetricsCapability" messageTypes="muws-xs1:ManagementEvent"/>
```

657

658 **mows-events:MetricsCapability** is a topic on which management events related to this
659 manageability capability SHOULD be emitted.

660

661 5.2.4 Operational State

662 This capability is identified by the following URI:

663 **<http://docs.oasis-open.org/wsdm/2004/12/mows/capabilities/OperationalState>**

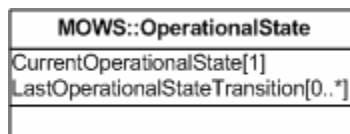
664 All properties, operations and events defined for this capability have the following metadata:

- 665 ▪ `<muws-xs2:Capability>http://docs.oasis-`
666 `open.org/wsdm/2004/12/mows/capabilities/OperationalState</muws-xs2:Capability>`

667

668 The Web service endpoint's manageable operational state capability is represented in the
669 **OperationalState** UML model class. The name of the class identifies the semantics of this
670 capability.

671



672

673 **Figure 10.** Endpoint operational state manageability capability model

674

675 The operational state model of a Web service endpoint used in this specification is the Web
676 service lifecycle (WSLC) state model as defined by the W3C Web Services Architecture
677 Management Task Force **[WSLC]**. Definition of the operational state in this specification uses the
678 transition paths for the service itself defined by the WSLC.

679

680 5.2.4.1 Information markup declarations

681 Each state MUST be identified by a QName and represented by a corresponding XML element.
682 Following is a list of elements corresponding to the operational states of the Web service
683 endpoint according to the WSLC state model **[WSLC]**.

- 684 ▪ **UpState**

685 This element corresponds to the WSLC UP top-level state which means that the Web
686 service endpoint is capable of accepting new requests. This element may contain the
687 BUSY and IDLE substates of UP, as defined below.

- 688 ▪ **DownState**

689 This element corresponds to the WSLC DOWN top-level state which means that the Web
690 service endpoint is not capable of accepting new requests. This element may contain the
691 STOPPED, CRASHED and SATURATED substates of DOWN, as defined below.

692 ▪ **BusyState**

693 This element corresponds to the WSLC BUSY substate of UP which means that the Web
694 service endpoint is capable of accepting new requests during processing of other
695 requests. This element MUST contain the UpState element.

696 ▪ **IdleState**

697 This element corresponds to the WSLC IDLE substate of UP which means that the Web
698 service endpoint is capable of accepting new requests and is not processing any other
699 requests. This element MUST contain the UpState element.

700 ▪ **StoppedState**

701 This element corresponds to the WSLC STOPPED substate of DOWN which means that
702 the Web service endpoint is not capable of accepting new requests and was intentionally
703 stopped by an administrator. This element MUST contain the DownState element.

704 ▪ **CrashedState**

705 This element corresponds to the WSLC CRASHED substate of DOWN which means that
706 the Web service endpoint is not capable of accepting new requests as a result of some
707 internal failure. This element MUST contain the DownState element

708 ▪ **SaturatedState**

709 This element corresponds to the WSLC SATURATED substate of DOWN which means
710 that the Web service endpoint is not capable of accepting new requests due to lack of
711 resources. This element MUST contain the DownState element.

712

713 It is possible to extend the above state model. Substates MAY be introduced and MUST be
714 identified by QNames, however, new top-level operational states MUST NOT be defined. In order
715 to represent the taxonomy lineage of substates in XML, the MUWS approach is used (§3.2 in the
716 **[MUWS]** part 2).

717

718 The **OperationalStateType** XML Schema type is declared as follows.

719

```
720 <xs:complexType name="OperationalStateType">  
721 <xs:complexContent>  
722 <xs:extension base="muws-xs2:StateType"/>  
723 </xs:complexContent>  
724 </xs:complexType>
```

725

726 The **OperationalStateType** is used to declare elements which contain any valid elements
727 designating an operational state of a Web service endpoint.

728

- 729 ▪ A substate of the operational state MUST be declared according to the following rules.
- 730 ○ An XML element is declared with a QName which identifies the desired substate
731 semantics, for example my-app:DatabaseCleanupState
 - 732 ○ The contents of the XML element MUST be the only element which corresponds
733 to the generalized state, for example mows-xs:StoppedState

734

735 An instance of the request processing state information represented in XML may look as shown
736 in the following example,

737

```
738 <my:OperationalStateInformationElement xsi:type="mows-xs:OperationalStateType">  
739   <my-app:DatabaseCleanupState>  
740     <mows-xs:StoppedState>  
741       <mows-xs:DownState/>  
742     </mows-xs:StoppedState>  
743   </my-app:DatabaseCleanupState>  
744 </my:RequestProcessingStateInformationElement>
```

745

746 5.2.4.2 Properties

747 The following is the specification of the Web service endpoint operational state properties (i.e. the
748 XML elements which represent the state properties).

749

```
750 <CurrentOperationalState>mows-xs:OperationalStateType</CurrentOperationalState>  
751 <LastOperationalStateTransition>  
752   muws-xs2:StateTransitionType  
753 </LastOperationalStateTransition> ?
```

754

755 **CurrentOperationalState** is the current operational state of the Web service endpoint being
756 managed. Metadata about this property is as follows.

- 757 ▪ Is *Mutable*
- 758 ▪ Is not *Modifiable*

759 **LastOperationalStateTransition** contains information about last operational state transition
760 which occurred at the Web service endpoint being managed. Metadata about this property is as
761 follows.

- 762 ▪ Is *Mutable*
- 763 ▪ Is not *Modifiable*

764

765 5.2.4.3 Events

766 The following specification defines this capability notification topics in the namespace mapped to
767 the **mows-events** prefix.

768

```
769 <wstop:Topic name="OperationalStateCapability" messageTypes="muws-  
770 xs1:ManagementEvent"/>
```

771

772 **mows-events:OperationalStateCapability** is a topic on which management events related to
773 this manageability capability SHOULD be emitted.

774

775 For information about changes of the operational state, a consumer MUST subscribe to
776 notifications on the changes of the CurrentOperationalState property (assuming that the
777 manageability endpoint implementation supports notifications about changes of this property).
778 Refer to **[WS-RP]** for information on how to subscribe to the property change notifications.

779

780 5.2.5 Operational Status

781 This capability is identified by the following URI:

782 <http://docs.oasis-open.org/wsdm/2004/12/mows/capabilities/OperationalStatus>

783 All properties, operations and events defined for this capability have the following metadata:

784 ▪ `<muws-xs2:Capability>http://docs.oasis-`
785 `open.org/wsdm/2004/12/mows/capabilities/OperationalStatus</muws-xs2:Capability>`

786

787 WSDM manageable endpoints that intend to support the MUWS **Operational Status**
788 manageability capability (§3.3 in the **[MUWS]** part 2) MUST abide by the following mapping rules.
789 When this capability support is indicated for a manageable endpoint, the mappings are in effect.

790

791 The Web service lifecycle (WSLC) states defined by the W3C Web Services Architecture
792 Management Task Force **[WSLC]** map to the MUWS status values as follows:

- 793 ▪ The WSLC **UP** state MUST be reported as the **Available** contents of the **muws-**
794 **xs2:OperationalStatus** property. Any sub-state of WSLC **UP** MUST be reported as
795 **Available**.
- 796 ▪ The WSLC **DOWN** state MUST be reported as the **Unavailable** contents of the **muws-**
797 **xs2:OperationalStatus** property. Any sub-state of WSLC **DOWN** SHOULD be reported
798 as **Unavailable**. The STOPPED and CRASHED substates of WSLC DOWN MUST be
799 reported as Unavailable.
- 800 ▪ The WSLC **SATURATED** sub-state of **DOWN** MAY be reported as the
801 **PartiallyAvailable** contents of the **muws-xs2:OperationalStatus** property.

802

803 5.2.5.1 Events

804 The following specification defines this capability notification topics in the namespace mapped to
805 the **mows-events** prefix.

806

```
807 <wstop:Topic name="OperationalStatusCapability" messageTypes="muws-  
808 xs1:ManagementEvent"/>
```

809

810 **mows-events:OperationalStatusCapability** is a topic on which management events related to
811 this manageability capability SHOULD be emitted.

812

813 5.2.6 Request Processing State

814 This capability is identified by the following URI:

815 <http://docs.oasis-open.org/wsdm/2004/12/mows/capabilities/RequestProcessingState>

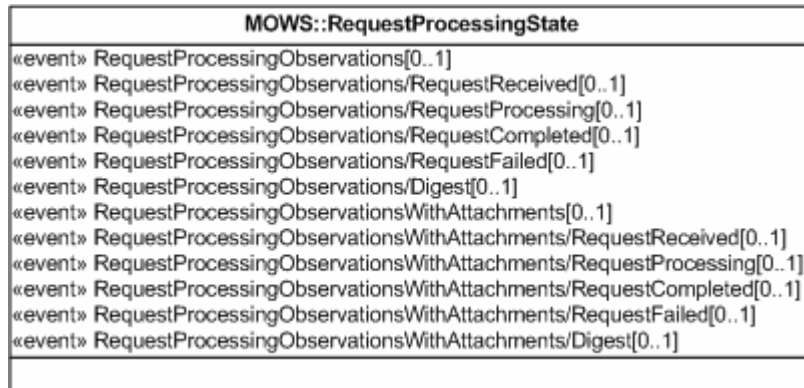
816 All properties, operations and events defined for this capability have the following metadata:

817 ▪ `<muws-xs2:Capability>http://docs.oasis-`
818 `open.org/wsdm/2004/12/mows/capabilities/RequestProcessingState</muws-`
819 `xs2:Capability>`

820

821 The Web service endpoint's manageable request processing state capability is represented in the
822 **RequestProcessingState** UML model class. The name of the class identifies the semantics of
823 this capability.

824



825

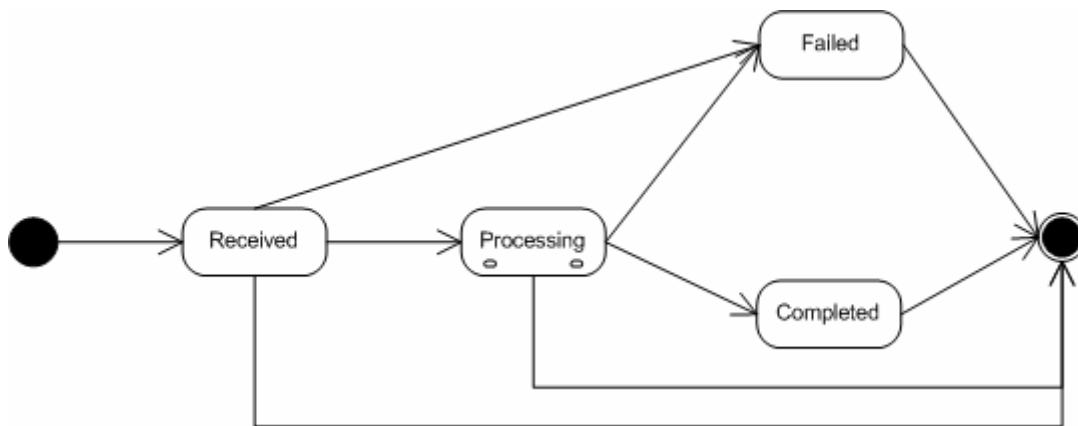
826

Figure 11. Endpoint request processing state manageability capability model

827

828 By the definition, a Web service endpoint accepts and processes messages targeted at it –
829 *requests*. Every request goes through a number of states (e.g. received, processing, completed
830 or failed) as defined by the [WSLC] and extended here.

831



832

833

Figure 12. Request processing states

834

835 The state diagram represents a model in which states MAY have duration and transitions are
836 instantaneous. When extending this model one MUST extend only the Processing compound
837 state.

838

839 5.2.6.1 Information markup declarations

840 Each state MUST be identified by a QName and represented by a corresponding XML element.
841 Following is a list of elements corresponding to the top-level states of the request processing
842 state model (Figure 12).

843

- **RequestReceivedState**

844

845

846

847

848

This element corresponds to the Received top-level state which means that the Web service endpoint has accepted a request to perform one of the service's functional responsibilities. This state represents the earliest point at which the manageability provider knows that the request was dispatched to the Web service endpoint being managed.

- 849 ▪ **RequestProcessingState**
- 850 This element corresponds to the Processing top-level state which means that the Web
- 851 service endpoint is doing some internal processing/execution to fulfill the requested
- 852 function. This state represents the earliest point at which the application module or
- 853 business logic begins processing the request. For example, if the application server
- 854 queues the request before dispatching it to the business logic, the time difference
- 855 between “request received” and “processing” will include the duration the request was
- 856 queued.
- 857 ▪ **RequestCompletedState**
- 858 This element corresponds to the Completed top-level state which means that the Web
- 859 service endpoint successfully completed requested function returning results to the
- 860 requester.
- 861 ▪ **RequestFailedState**
- 862 This element corresponds to the Failed top-level state which means that the Web service
- 863 endpoint encountered an error and didn't complete the requested function, returning
- 864 error/fault to the requester.

865

866 It is possible to extend the above state model. Substates of the Processing top-level state MAY
867 be introduced and MUST be identified by QNames, however, new top-level request processing
868 states MUST NOT be defined. In order to represent the taxonomy lineage of substates in XML,
869 the MUWS approach is used (§3.2 in the [MUWS] part 2).

870

871 The **RequestProcessingStateType** XML Schema type is declared as follows.

872

```
873 <xs:complexType name="RequestProcessingStateType">
874 <xs:complexContent>
875     <xs:extension base="muws-xs2:StateType"/>
876 </xs:complexContent>
877 </xs:complexType>
```

878

879 The **RequestProcessingStateType** is used to declare elements which designate a request
880 processing state – top-level or substates of the Processing.

881

882 A substate of the Processing compound state MUST be declared according to the following rules.

883 An XML element is declared with a QName which identifies the desired substate semantics, for
884 example my-soap:SerializationState

885 The contents of the XML element MUST be the only element which corresponds to the
886 generalized state, for example muws-xs2:RequestProcessingState

887

888 An instance of the request processing state information represented in XML may appear as
889 shown in the following example,

890

```
891 <my:RequestProcessingStateInformationElement xsi:type="mows-
892 xs:RequestProcessingStateType">
893     <my-soap:SerializationState>
894         <mows-xs:RequestProcessingState/>
895     </my-soap:SerializationState>
896 </my:RequestProcessingStateInformationElement>
```

897

898 5.2.6.2 Events

899 Notifications are emitted when requests enter one of the request processing states (Figure 12).

900

901 The following specification defines the Web service endpoint request processing state notification
902 topics in the namespace mapped to the **mows-events** prefix. The message patterns' expression
903 and dialect MUST match precisely what is declared below.

904

```
905 <wstop:Topic name="RequestProcessingStateCapability" messageTypes="muws-
906 xs1:ManagementEvent"/>
907
908 <wstop:Topic name="RequestProcessingObservations"
909     messageTypes="muws-xs1:ManagementEvent">
910     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
911 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
912 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
913 xs:RequestProcessingNotification)=1]
914     </wstop:MessagePattern>
915     <wstop:Topic name="RequestReceived"
916         messageTypes="muws-xs1:ManagementEvent">
917         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
918 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
919 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
920 xs:RequestProcessingNotification)=1]
921         </wstop:MessagePattern>
922     </wstop:Topic>
923     <wstop:Topic name="RequestProcessing"
924         messageTypes="muws-xs1:ManagementEvent">
925         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
926 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
927 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
928 xs:RequestProcessingNotification)=1]
929         </wstop:MessagePattern>
930     </wstop:Topic>
931     <wstop:Topic name="RequestCompleted"
932         messageTypes="muws-xs1:ManagementEvent">
933         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
934 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
935 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
936 xs:RequestProcessingNotification)=1]
937         </wstop:MessagePattern>
938     </wstop:Topic>
939     <wstop:Topic name="RequestFailed"
940         messageTypes="muws-xs1:ManagementEvent">
941         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
942 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
943 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
944 xs:RequestProcessingNotification)=1]
945         </wstop:MessagePattern>
946     </wstop:Topic>
947     <wstop:Topic name="Digest"
948         messageTypes="muws-xs1:ManagementEvent">
949     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
```

```

950 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
951 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
952 xs:RequestProcessingNotification)=1]
953     </wstop:MessagePattern>
954     </wstop:Topic>
955 </wstop:Topic>
956
957 <wstop:Topic name="RequestProcessingObservationsWithAttachments"
958     messageTypes="muws-xs1:ManagementEvent">
959     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
960 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
961 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
962 xs:RequestProcessingNotification)=1]
963     </wstop:MessagePattern>
964     <wstop:Topic name="RequestReceived"
965         messageTypes="muws-xs1:ManagementEvent">
966         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
967 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
968 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
969 xs:RequestProcessingNotification)=1]
970         </wstop:MessagePattern>
971         </wstop:Topic>
972         <wstop:Topic name="RequestProcessing"
973             messageTypes="muws-xs1:ManagementEvent">
974             <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
975 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
976 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
977 xs:RequestProcessingNotification)=1]
978             </wstop:MessagePattern>
979             </wstop:Topic>
980             <wstop:Topic name="RequestCompleted"
981                 messageTypes="muws-xs1:ManagementEvent">
982                 <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
983 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
984 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
985 xs:RequestProcessingNotification)=1]
986                 </wstop:MessagePattern>
987                 </wstop:Topic>
988                 <wstop:Topic name="RequestFailed"
989                     messageTypes="muws-xs1:ManagementEvent">
990                     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
991 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
992 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
993 xs:RequestProcessingNotification)=1]
994                     </wstop:MessagePattern>
995                     </wstop:Topic>
996                     <wstop:Topic name="Digest"
997                         messageTypes="muws-xs1:ManagementEvent">
998                         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
999 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1000 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1001 xs:RequestProcessingNotification)=1]
1002                         </wstop:MessagePattern>
1003                         </wstop:Topic>
1004                     </wstop:Topic>

```

1006 **mows-events:RequestProcessingStateCapability** is a topic on which management events
1007 related to this manageability capability SHOULD be emitted.

1008 **mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations**
1009 indicates availability of any information about the processing of any request by the Web service
1010 endpoint (Figure 12) as observed by the implementation of a manageable Web service.

1011 The notification message for this topic MUST contain at most one
1012 **RequestProcessingNotification** element (defined in §5.2.6.2.1). The MUWS
1013 management event MUST also declare the event situation category with the muws-
1014 xs2:ReportSituation element and the severity value “1” (Informational). It is recommended
1015 to subscribe to these topics with proper preconditions and selectors using expressions
1016 against the contents of the RequestProcessingNotification element.

1017 **mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations/mows-**
1018 **events:RequestReceived** indicates that a request was received by the Web service endpoint
1019 being managed (Received state on the Figure 12). The notification message format for this topic
1020 is the same as the notification message format for the mows-events:ManageableEndpoint/mows-
1021 events:RequestProcessingObservations topic. This is a state change event and therefore
1022 notification messages MUST contain exactly one muws-xs2:StateTransition element inside of the
1023 RequestProcessingNotification/StateInformation element.

1024 **mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations/mows-**
1025 **events:RequestProcessing** indicates that a request is being processed by the Web service
1026 endpoint being managed (Processing state on the Figure 12). The notification message format for
1027 this topic is the same as the notification message format for the mows-
1028 events:ManageableEndpoint/mows-events:RequestProcessingObservations topic. This is a state
1029 change event and therefore notification messages MUST contain exactly one muws-
1030 xs2:StateTransition element inside of the RequestProcessingNotification/StateInformation
1031 element.

1032 **mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations/mows-**
1033 **events:RequestCompleted** indicates that a request was successfully completed by the Web
1034 service endpoint being managed (Completed state on the Figure 12). The notification message
1035 format for this topic is the same as the notification message format for the mows-
1036 events:ManageableEndpoint/mows-events:RequestProcessingObservations topic. This is a state
1037 change event and therefore notification messages MUST contain exactly one muws-
1038 xs2:StateTransition element inside of the RequestProcessingNotification/StateInformation
1039 element.

1040 **mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations/mows-**
1041 **events:RequestFailed** indicates that a request was failed (not successfully completed) by the
1042 Web service endpoint being managed (Failed state on the Figure 12). The notification message
1043 format for this topic is the same as the notification message format for the mows-
1044 events:ManageableEndpoint/mows-events:RequestProcessingObservations topic. This is a state
1045 change event and therefore notification messages MUST contain exactly one muws-
1046 xs2:StateTransition element inside of the RequestProcessingNotification/StateInformation
1047 element.

1048 **mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations/mows-**
1049 **events:Digest** indicates availability of summary information about a request processed by the
1050 Web service endpoint being managed. The notification message format for this topic is the same
1051 as the notification message format for the mows-events:ManageableEndpoint/mows-
1052 events:RequestProcessingObservations topic. This is a digest event and therefore notification
1053 messages MUST contain one or more muws-xs2:StateTransition elements inside of the
1054 RequestProcessingNotification/StateInformation element. Each muws-xs2:StateTransition
1055 element describes a state transition which occurred with that one request which this summary
1056 notification is informing about. Each state transition information element carries an attribute
1057 indicating the time when that particular transition occurred. Using this information the
1058 manageability consumer can reconstruct the sequence of events with regards to the request.

1059 **mows-events:ManageableEndpoint/mows-**
1060 **events:RequestProcessingObservationsWithAttachments** topic and all of its subtopics are
1061 defined exactly as the mows-events:ManageableEndpoint/mows-
1062 events:RequestProcessingObservations topic and its respective subtopics, except that the
1063 notification messages MUST include attachments (if any) of the request and reply messages sent
1064 to/from the Web service endpoint being managed.

1065 The notification message format for this topic and all of its subtopics is the same as the
1066 notification message format for the mows-events:ManageableEndpoint/mows-
1067 events:RequestProcessingObservations topic, except that attachments may be sent
1068 along with the message. The precise mechanism of sending the attachment is dependent
1069 on 1) the binding of the notification consumer Web service endpoint **[WS-N]** and 2) the
1070 binding of the Web service endpoint being managed.

1071

1072 The mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations/mows-
1073 events:RequestProcessing topic MAY be extended with custom subtopics in order to represent
1074 custom request processing substates of the Processing compound state (Figure 12).

1075

1076 Note that the result of the message pattern XPath expressions in the topic declarations above is
1077 the XML nodeset **[XPath]** of the notification messages that are sent inside of the S:Body element
1078 or the wsnt:Notify element **[WS-N]**.

1079

1080 Note that for the XPath expressions defined here the prefix-to-namespace mapping context
1081 MUST include all prefixes which appear in the XPath expression and mapped according to the
1082 table in the §5.

1083

1084 **5.2.6.2.1 RequestProcessingNotification message**

1085 The RequestProcessingNotification message format is defined as follows.

1086

```
1087 <RequestProcessingNotification CurrentTime="xs:dateTime" ...>
1088 <Request ...>
1089   <TransportInformation ...> {any}* </TransportInformation> ?
1090   <Message ...>
1091     <Size Unit=("bit" | "byte" | "word" | "dword" | "qword")
1092       ...>xs:positiveInteger</Size> ?
1093     (
1094       <NotIncluded/> |
1095       <Text>xs:string</Text> |
1096       <Binary>xs:base64Binary</Binary> |
1097       <Xml>{any}*</Xml>
1098     )
1099     {any}*
1100   </Message>
1101   {any}*
1102 </Request> ?
1103 <Reply ...>
1104   <!-- ... see contents of the Request element above ... -->
1105 </Reply> ?
1106 <StateInformation>
1107 <muws-xs2:StateTransition> <!-- ...see [MUWS]... --> </muws-xs2:StateTransition> +
1108 </StateInformation>
1109 {any}*
```

1110 </RequestProcessingNotification>

1111

1112 **RequestProcessingNotification** is a container element of the information about a request going
1113 through the request processing states (Figure 12).

1114 **RequestProcessingNotification/@CurrentTime** indicates current time measured at the
1115 manageability endpoint. All time/date values in this notification information are synchronized with
1116 this time indication.

1117 **RequestProcessingNotification/Request** element contains information about the request itself.
1118 Note that the request is not necessarily serialized as a SOAP message. Therefore, the contents
1119 allow information about requests in general, however the information has to be serializable in
1120 XML [XML]. The presence of this element in the notification MUST indicate presence of the
1121 actual request message sent to the Web service endpoint being managed. The contents may
1122 vary depending on what the implementation of the manageability endpoint can or intends to
1123 provide. For example, for security reasons the actual contents of the message may be omitted.
1124 However, in order to indicate that the request message exists, this element has to be included in
1125 the notification.

1126 **RequestProcessingNotification/Request/TransportInformation** element contains information
1127 about the transport by which the request was received. The content of this element is open, but
1128 WSDM defines the following elements useable for TCP/IP transports.

```
1129 <TcpIpInfo
1130     Direction=("from" | "to")
1131     Port="xs:positiveInteger"
1132     Protocol=("TCP" | "UDP") ...>
1133     (
1134     <IPV4Address>
1135         xs:hexBinary[xs:length[@value="8" and @fixed="true"]]
1136     </IPV4Address> |
1137     <IPV6Address>
1138         xs:hexBinary[xs:length[@value="32" and @fixed="true"]]
1139     </IPV6Address>
1140     )
1141     {any}*
1142 </TcpIpInfo>
```

1143 **TcpIpInfo** contains information about a communication to or from an IP addressable
1144 network device.

1145 **TcpIpInfo/@Direction** indicates communication to or from the IP addressable network
1146 device.

1147 **TcpIpInfo/@Port** is a TCP/IP network port number used on the IP addressable network
1148 device.

1149 **TcpIpInfo/@Protocol** indicates if the TCP or UDP protocol is used.

1150 **TcpIpInfo/IPV4Address** contains hexadecimal representation of the IP address version
1151 4. The value MUST represent 32 bits.

1152 **TcpIpInfo/IPV6Address** contains hexadecimal representation of the IP address version
1153 6. The value MUST represent 128 bits.

1154 **RequestProcessingNotification/Request/Message** element contains the message observed by
1155 the Web service endpoint being managed.

1156 **RequestProcessingNotification/Request/Message/Size** indicates size of the message. When
1157 subscribed to observations with attachments, this value includes the size of the message payload
1158 plus all the attachments. Otherwise, just the payload of the message (i.e. size of the contents of
1159 the RequestProcessingNotification/Request/Message element) is reported. Note that the actual
1160 message contents may not be reported for security reasons, however size may be reported.

1161 **RequestProcessingNotification/Request/Message/Size/@Unit** indicates what units were used
1162 to calculate the size of the message. The valid values of this attribute are:

1163 **bit** – size indicates number of bits in the message.
1164 **byte** – size indicates number of bytes (8 bit sets) in the message
1165 **word** – size indicates number of double bytes (16 bit sets) in the message.
1166 **dword** – size indicates number of double words (32 bit sets) in the message.
1167 **qword** – size indicates number of quad words (64 bit sets) in the message.

1168 **RequestProcessingNotification/Request/Message/NotIncluded** element indicates that the
1169 message content is intentionally not provided by the implementation of the Web service endpoint
1170 manageability.

1171 **RequestProcessingNotification/Request/Message/Text** element contains the observed
1172 message's text representation. For example, a non-well formed XML message should be
1173 represented as text. It is recommended that text data is wrapped in an XML CDATA section
1174 **[XML]**.

1175 **RequestProcessingNotification/Request/Message/Binary** element contains the binary
1176 representation of the observed message. If a message cannot be represented as either well-
1177 formed XML nor as text, it should be binary encoded.

1178 **RequestProcessingNotification/Request/Message/XML** element contains the observed
1179 message's XML representation. For example, a SOAP message envelope element (S:Envelope)
1180 may appear in the contents.

1181 **RequestProcessingNotification/Request/{any}** is an extensibility element where additional
1182 information about the request MAY appear. The form of the information representation in XML is
1183 manageability endpoint implementation specific. In other words, vendor extensions may appear
1184 here.

1185 The **RequestProcessingNotification/Reply** element contains information about the reply (if any)
1186 for the request. Note that fault is also a valid reply element. The content of this element has the
1187 same format as the RequestProcessingNotification/Request element.

1188 **RequestProcessingNotification/StateInformation** element contains information about the
1189 request processing state.

1190 **RequestProcessingNotification/StateInformation/muws-xs2:StateTransition** element
1191 contains information about a state transition. There MUST be exactly one such element for each
1192 state change event. There MUST be one or more such elements for the digest event.

1193 **RequestProcessingNotification/StateInformation/muws-xs2:StateTransition/@muws-**
1194 **xs2:Time** indicates time when the described transition occurred. Note that according to the
1195 request processing state model (Figure 12), all transitions are instantaneous. Time is measured
1196 at the implementation of the manageability endpoint and is synchronized with the
1197 RequestProcessingNotification/@CurrentTime value reading.

1198 **RequestProcessingNotification/StateInformation/muws-xs2:StateTransition/muws-**
1199 **xs2:EnteredState** indicates which request processing state was entered.

1200 **RequestProcessingNotification/StateInformation/muws-xs2:StateTransition/muws-**
1201 **xs2:PreviousState** indicates which request processing state was exited.

1202 **RequestProcessingNotification/{any}** is an extensibility element where additional information
1203 about this request processing notification MAY appear. The form of the information representation
1204 in XML is manageability endpoint implementation specific. In other words, vendor extensions may
1205 appear here.

1206 The contents of the RequestProcessingNotification element SHOULD be used to specify
1207 selectors **[WS-N]** when subscribing to notification messages containing this element.

5.2.6.2.2 Examples of events against the Web service endpoint request processing state

Consider the following message exchange with a fictitious order-entry Web service endpoint.

Request:

```
<S:Envelope xmlns:x="..." ... >
...
<S:Body>
  <x:Order>
    <x:Item>...</x:Item>
    <x:Quantity>...</x:Quantity>
  </x:Order>
</S:Body>
</S:Envelope>
```

Reply:

```
<S:Envelope xmlns:x="..." ... >
...
<S:Body>
  <x:Shipped>
    <x:Item>...</x:Item>
    <x:Quantity>...</x:Quantity>
  </x:Shipped>
</S:Body>
</S:Envelope>
```

To be notified of a particular item shortage when the order request is processed and the shipped quantity is less than the ordered quantity, the following XPath selector should be specified when subscribing to the **mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations/mows-events:RequestCompleted** topic.

Selector:

```
boolean(//mows-xs:RequestProcessingNotification[mows-xs:Request/mows-xs:Message/mows-
xs:Xml//x:Order/x:Quantity < mows-xs:Reply/mows-xs:Message/mows-
xs:Xml//x:Shipped/x:Quantity])
```

This way, when the condition is met, the manageable Web service endpoint will emit the notification message containing the **RequestProcessingNotification** element with the following contents.

```
<RequestProcessingNotification CurrentTime="...">
<Request>
  <TransportInformation>
    <TcpIpInfo Direction="from" Port="2840" Protocol="TCP">
      <IPV4Address>C0A80002</IPV4Address>
    </TcpIpInfo>
    <TcpIpInfo Direction="to" Port="80" Protocol="TCP">
      <IPV4Address>C0A80003</IPV4Address>
    </TcpIpInfo>
  </TransportInformation>
```

```

1258     <Message>
1259     <Size Unit="byte">257</Size>
1260     <Xml>
1261     <S:Envelope xmlns:S="..." xmlns:x="..." ...>
1262     . . .
1263     <S:Body>
1264         <x:Order>
1265             <x:Item>123</x:Item>
1266             <x:Quantity>10</x:Quantity>
1267         </x:Order>
1268     </S:Body>
1269 </S:Envelope>
1270 </Xml>
1271 </Message>
1272 </Request>
1273 <Reply>
1274     <TransportInformation>
1275         <TcplpInfo Direction="to" Port="2840" Protocol="TCP">
1276             <IPV4Address>C0A80002</IPV4Address>
1277         </TcplpInfo>
1278         <TcplpInfo Direction="from" Port="80" Protocol="TCP">
1279             <IPV4Address>C0A80003</IPV4Address>
1280         </TcplpInfo>
1281     </TransportInformation>
1282     <Message>
1283     <Size Unit="byte">232</Size>
1284     <Xml>
1285     <S:Envelope xmlns:S="..." xmlns:x="..." ...>
1286     . . .
1287     <S:Body>
1288         <x:Shipped>
1289             <x:Item>123</x:Item>
1290             <x:Quantity>2</x:Quantity>
1291         </x:Shipped>
1292     </S:Body>
1293 </S:Envelope>
1294 </Xml>
1295 </Message>
1296 </Reply>
1297 <muws-xs2:StateTransition Time="...">
1298 <muws-xs2:EnteredState/><RequestCompletedState/></muws-xs2:EnteredState>
1299 <muws-xs2:PreviousState><RequestProcessingState/></muws-xs2:PreviousState>
1300 </muws-xs2:StateTransition>
1301 . . .
1302 </RequestProcessingNotification>
1303
1304

```

1305

6 References

1306

6.1 Normative

- 1307 [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
1308 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 1309 [MUWS] William Vambenepe, *Web Services Distributed*
1310 *Management:Management using Web Services (MUWS 1.0) Part 1*,
1311 OASIS Committee Draft, December 2004, [http://docs.oasis-
1313 open.org/wsdm/2004/12/cd-wsdm-muws-part1-1.0.pdf](http://docs.oasis-
1312 open.org/wsdm/2004/12/cd-wsdm-muws-part1-1.0.pdf)
1314 William Vambenepe, *Web Services Distributed*
1315 *Management:Management using Web Services (MUWS 1.0) Part 2*,
1316 OASIS Committee Draft, December 2004, [http://docs.oasis-
open.org/wsdm/2004/12/cd-wsdm-muws-part2-1.0.pdf](http://docs.oasis-
open.org/wsdm/2004/12/cd-wsdm-muws-part2-1.0.pdf)
- 1317 [WS-A] Don Box, et al., *Web services Addressing (WS-Addressing)*, W3C
1318 Member Submission, August 2004,
1319 <http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/>
- 1320 [WS-RP] Steve Graham, et al., *Web Services Resource Properties 1.2 (WS-
1321 ResourceProperties)*, OASIS Working Draft, June 2004,
1322 [http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-
1.2-draft-04.pdf](http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-
1323 1.2-draft-04.pdf)
- 1324 [WS-N] Steve Graham, et al., *Web Services Base Notification 1.2 (WS-
1325 BaseNotification)*, OASIS Working Draft, June 2004, [http://docs.oasis-
open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf](http://docs.oasis-
1326 open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf)
- 1327 [WS-T] William Vambenepe, *Web Services Topics 1.2 (WS-Topics)*, OASIS
1328 Working Draft, July 2004, [http://docs.oasis-open.org/wsn/2004/06/wsn-
WS-Topics-1.2-draft-01.pdf](http://docs.oasis-open.org/wsn/2004/06/wsn-
1329 WS-Topics-1.2-draft-01.pdf)
- 1330 [WSDL] Erik Christensen, et al., *Web services Description Language (WSDL)*
1331 *1.1*, W3C Note, March 2001, <http://www.w3.org/TR/wSDL>
- 1332 [SOAP] Don Box, et al., *Simple Object Access Protocol 1.1*, W3C Note, May
1333 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- 1334 [XMLS] Henry S. Thompson, et al. *XML Schema Part 1: Structures*, W3C
1335 Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-1/>
1336 Paul V. Biron, et al. *XML Schema Part 2: Datatypes*, W3C
1337 Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-2/>
- 1338 [XML] Tim Bray, et al., *Extensible Markup Language (XML) 1.0 (Third Edition)*,
1339 W3C Recommendation, February 2004, <http://www.w3.org/TR/REC-xml>
- 1340 [XNS] Tim Bray, et al., *Namespaces in XML*, W3C Recommendation, January
1341 1999, <http://www.w3.org/TR/REC-xml-names/>
- 1342 [XPath] James Clark, et al., *XML Path Language (XPath) Version 1.0*, W3C
1343 Recommendation, November 1999, [http://www.w3.org/TR/1999/REC-
xpath-19991116](http://www.w3.org/TR/1999/REC-
1344 xpath-19991116)

1345

6.2 Non-normative

- 1346 [MOWS-Reqs] Mark Potts, et al., *WSDM Management of Web Services Requirements*,
1347 October 2003, [http://www.oasis-
open.org/apps/org/workgroup/wsdm/download.php/3887/WSDM-MOWS-
Requirements.20031008.doc](http://www.oasis-
1348 open.org/apps/org/workgroup/wsdm/download.php/3887/WSDM-MOWS-
1349 Requirements.20031008.doc)

1350	[WS-Arch]	David Booth, et al. <i>Web Services Architecture</i> , W3C Working Group Note, February 2004, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/
1351		
1352		
1353	[WSLC]	Hao He, et al., <i>Web Service Management: Service Lifecycle</i> , W3C Note, February 2004, http://www.w3.org/TR/2004/NOTE-wslc-20040211/
1354		
1355	[WSS]	Anthony Nadalin, et al., <i>Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)</i> , March 2004, OASIS Standard, http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf
1356		
1357		
1358	[BSP]	Abbie Barbir, et al., <i>Basic Security Profile Version 1.0</i> , WS-I Working Group Draft, May 2004, http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0-2004-05-12.html
1359		
1360		
1361		

1362

Appendix A. Acknowledgments

1363 The following people made contributions to this specification: Brian Carol, Fred Carter, John
1364 DeCarlo, Andreas Dharmawan, Maryann Hondo, Heather Kreger, Bryan Murray, Micheal Perks,
1365 Igor Sedukhin, William Vambenepe, Andrea Westerinen.

1366

1367 The following individuals were members of the committee when the specification was approved
1368 by the technical committee: Guru Bhat, Jeff Bohren, Winston Bumpus, Nick Butler, Brian Carroll,
1369 Fred Carter, Michael Clements, David Cox, John DeCarlo, Andreas Dharmawan, Mark Ellison,
1370 John Fuller, Paul Lipton, Heather Kreger, Hal Lockhart, Frederico Maciel, Tom Maguire, Bryan
1371 Murray, Richard Nikula, Mark Peel, Richard Pelavin, Hodayoun Pourheidari, Warren Roberts,
1372 Karl Schopmeyer, Igor Sedukhin, David Snelling, Thomas Studwell, William Vambenepe, Andrea
1373 Westerinen, Jim Willits, Zhili Zhang.

Appendix B. Revision History

Rev	Date	By Whom
wd	2003-10-31	Igor Sedukhin
wd	2003-11-14	Igor Sedukhin
wd	2003-12-02	Igor Sedukhin
wd	2004-01-26	Igor Sedukhin
wd	2004-02-17	Igor Sedukhin
wd	2004-03-01	Igor Sedukhin
wd	2004-03-18	Igor Sedukhin
wd	2004-03-19	Igor Sedukhin
wd	2004-03-24	Igor Sedukhin
wd	2004-03-24	Igor Sedukhin
cd	2004-04-02	Igor Sedukhin
wd	2004-07-21	Igor Sedukhin
wd	2004-09-11	Igor Sedukhin
wd	2004-10-11	Igor Sedukhin
wd	2004-10-24	Igor Sedukhin
wd	2004-11-04	Igor Sedukhin
wd	2004-11-15	Igor Sedukhin
wd	2004-11-19	Igor Sedukhin
wd	2004-11-23	Igor Sedukhin
wd	2004-12-03	Igor Sedukhin
cd	2004-12-10	Igor Sedukhin

1376

Appendix C. Notices

1377 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
1378 that might be claimed to pertain to the implementation or use of the technology described in this
1379 document or the extent to which any license under such rights might or might not be available;
1380 neither does it represent that it has made any effort to identify any such rights. Information on
1381 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
1382 website. Copies of claims of rights made available for publication and any assurances of licenses
1383 to be made available, or the result of an attempt made to obtain a general license or permission
1384 for the use of such proprietary rights by implementors or users of this specification, can be
1385 obtained from the OASIS Executive Director.

1386 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
1387 applications, or other proprietary rights which may cover technology that may be required to
1388 implement this specification. Please address the information to the OASIS Executive Director.

1389 Copyright © OASIS Open 2003-2004. *All Rights Reserved.*

1390 This document and translations of it may be copied and furnished to others, and derivative works
1391 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
1392 published and distributed, in whole or in part, without restriction of any kind, provided that the
1393 above copyright notice and this paragraph are included on all such copies and derivative works.
1394 However, this document itself does not be modified in any way, such as by removing the
1395 copyright notice or references to OASIS, except as needed for the purpose of developing OASIS
1396 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
1397 Property Rights document must be followed, or as required to translate it into languages other
1398 than English.

1399 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
1400 successors or assigns.

1401 This document and the information contained herein is provided on an "AS IS" basis and OASIS
1402 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
1403 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
1404 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
1405 PARTICULAR PURPOSE.

1406

Appendix D. XML Schemas

```

1408 <?xml version="1.0" encoding="utf-8"?>
1409 <xs:schema targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-
1410 mows.xsd" xmlns:mows-xs="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.xsd"
1411 xmlns:muws-xs2="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part2.xsd"
1412 xmlns:muws-xs1="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part1.xsd"
1413 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1414 xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
1415 attributeFormDefault="unqualified">
1416
1417     <xs:import namespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1418 part1.xsd" schemaLocation="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1419 part1.xsd"/>
1420     <xs:import namespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1421 part2.xsd" schemaLocation="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1422 part2.xsd"/>
1423     <xs:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1424 schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
1425
1426     <!-- MOWS::ManageabilityReferences -->
1427     <xs:element name="GetManageabilityReferences"/>
1428     <xs:element name="GetManageabilityReferencesResponse">
1429         <xs:complexType>
1430             <xs:sequence>
1431                 <xs:element ref="muws-xs1:ManageabilityEndpointReference"
1432                     maxOccurs="unbounded"/>
1433             </xs:sequence>
1434         </xs:complexType>
1435     </xs:element>
1436
1437     <!-- MOWS::Identification -->
1438     <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
1439     <xs:element name="EndpointDescriptions">
1440         <xs:complexType>
1441             <xs:sequence>
1442                 <xs:element name="description" type="xs:anyURI"
1443                     minOccurs="0" maxOccurs="unbounded"/>
1444             </xs:sequence>
1445             <xs:anyAttribute namespace="##other" processContents="lax"/>
1446         </xs:complexType>
1447     </xs:element>
1448
1449     <xs:complexType name="EndpointIdentificationPropertiesType">
1450         <xs:sequence>
1451             <xs:element ref="mows-xs:EndpointReference"/>
1452             <xs:element ref="mows-xs:EndpointDescriptions" minOccurs="0"/>
1453         </xs:sequence>
1454     </xs:complexType>
1455     <xs:element name="EndpointIdentificationProperties"
1456         type="mows-xs:EndpointIdentificationPropertiesType"/>
1457
1458     <!-- MOWS::Metrics -->

```



```

1459 <xs:complexType name="IntegerCounter">
1460     <xs:simpleContent>
1461         <xs:extension base="xs:nonNegativeInteger">
1462             <xs:attributeGroup ref="muws-xs2:MetricAttributes"/>
1463             <xs:anyAttribute namespace="##other" processContents="lax"/>
1464         </xs:extension>
1465     </xs:simpleContent>
1466 </xs:complexType>
1467
1468 <xs:complexType name="DurationMetric">
1469     <xs:simpleContent>
1470         <xs:extension base="xs:duration">
1471             <xs:attributeGroup ref="muws-xs2:MetricAttributes"/>
1472             <xs:anyAttribute namespace="##other" processContents="lax"/>
1473         </xs:extension>
1474     </xs:simpleContent>
1475 </xs:complexType>
1476
1477 <xs:element name="NumberOfRequests" type="mows-xs:IntegerCounter"/>
1478 <xs:element name="NumberOfSuccessfulRequests" type="mows-xs:IntegerCounter"/>
1479 <xs:element name="NumberOfFailedRequests" type="mows-xs:IntegerCounter"/>
1480 <xs:element name="ServiceTime" type="mows-xs:DurationMetric"/>
1481 <xs:element name="MaxResponseTime" type="mows-xs:DurationMetric"/>
1482 <xs:element name="LastResponseTime" type="mows-xs:DurationMetric"/>
1483
1484 <xs:complexType name="EndpointMetricsPropertiesType">
1485     <xs:sequence>
1486         <xs:element ref="mows-xs:NumberOfRequests" minOccurs="0"/>
1487         <xs:element ref="mows-xs:NumberOfFailedRequests" minOccurs="0"/>
1488         <xs:element ref="mows-xs:NumberOfSuccessfulRequests"
1489             minOccurs="0"/>
1490         <xs:element ref="mows-xs:ServiceTime" minOccurs="0"/>
1491         <xs:element ref="mows-xs:MaxResponseTime" minOccurs="0"/>
1492         <xs:element ref="mows-xs:LastResponseTime" minOccurs="0"/>
1493     </xs:sequence>
1494 </xs:complexType>
1495 <xs:element name="EndpointMetricsProperties"
1496     type="mows-xs:EndpointMetricsPropertiesType"/>
1497
1498 <!-- MOWS::OperationalState -->
1499 <xs:complexType name="OperationalStateType">
1500     <xs:complexContent>
1501         <xs:extension base="muws-xs2:StateType"/>
1502     </xs:complexContent>
1503 </xs:complexType>
1504 <xs:element name="UpState">
1505     <xs:complexType>
1506         <xs:complexContent>
1507             <xs:restriction base="mows-xs:OperationalStateType"/>
1508         </xs:complexContent>
1509     </xs:complexType>
1510 </xs:element>
1511 <xs:element name="IdleState">
1512     <xs:complexType>
1513         <xs:complexContent>
1514             <xs:restriction base="mows-xs:OperationalStateType">

```

```

1515         <xs:sequence>
1516             <xs:element ref="mows-xs:UpState"/>
1517         </xs:sequence>
1518     </xs:restriction>
1519 </xs:complexContent>
1520 </xs:complexType>
1521 </xs:element>
1522 <xs:element name="BusyState">
1523     <xs:complexType>
1524         <xs:complexContent>
1525             <xs:restriction base="mows-xs:OperationalStateType">
1526                 <xs:sequence>
1527                     <xs:element ref="mows-xs:UpState"/>
1528                 </xs:sequence>
1529             </xs:restriction>
1530         </xs:complexContent>
1531     </xs:complexType>
1532 </xs:element>
1533 <xs:element name="DownState">
1534     <xs:complexType>
1535         <xs:complexContent>
1536             <xs:restriction base="mows-xs:OperationalStateType"/>
1537         </xs:complexContent>
1538     </xs:complexType>
1539 </xs:element>
1540 <xs:element name="StoppedState">
1541     <xs:complexType>
1542         <xs:complexContent>
1543             <xs:restriction base="mows-xs:OperationalStateType">
1544                 <xs:sequence>
1545                     <xs:element ref="mows-xs:DownState"/>
1546                 </xs:sequence>
1547             </xs:restriction>
1548         </xs:complexContent>
1549     </xs:complexType>
1550 </xs:element>
1551 <xs:element name="CrashedState">
1552     <xs:complexType>
1553         <xs:complexContent>
1554             <xs:restriction base="mows-xs:OperationalStateType">
1555                 <xs:sequence>
1556                     <xs:element ref="mows-xs:DownState"/>
1557                 </xs:sequence>
1558             </xs:restriction>
1559         </xs:complexContent>
1560     </xs:complexType>
1561 </xs:element>
1562 <xs:element name="SaturatedState">
1563     <xs:complexType>
1564         <xs:complexContent>
1565             <xs:restriction base="mows-xs:OperationalStateType">
1566                 <xs:sequence>
1567                     <xs:element ref="mows-xs:DownState"/>
1568                 </xs:sequence>
1569             </xs:restriction>
1570         </xs:complexContent>

```

```

1571         </xs:complexType>
1572     </xs:element>
1573
1574     <xs:element name="CurrentOperationalState" type="mows-xs:OperationalStateType"/>
1575     <xs:element name="LastOperationalStateTransition"
1576         type="muws-xs2:StateTransitionType"/>
1577
1578     <xs:complexType name="EndpointOperationalStatePropertiesType">
1579         <xs:sequence>
1580             <xs:element ref="mows-xs:CurrentOperationalState"/>
1581             <xs:element ref="mows-xs:LastOperationalStateTransition"
1582                 minOccurs="0"/>
1583         </xs:sequence>
1584     </xs:complexType>
1585     <xs:element name="EndpointOperationalStateProperties"
1586         type="mows-xs:EndpointOperationalStatePropertiesType"/>
1587
1588     <!-- MOWS::RequestProcessingState -->
1589     <xs:complexType name="RequestProcessingStateType">
1590         <xs:complexContent>
1591             <xs:extension base="muws-xs2:StateType"/>
1592         </xs:complexContent>
1593     </xs:complexType>
1594     <xs:element name="RequestReceivedState">
1595         <xs:complexType>
1596             <xs:complexContent>
1597                 <xs:restriction base="mows-xs:RequestProcessingStateType"/>
1598             </xs:complexContent>
1599         </xs:complexType>
1600     </xs:element>
1601     <xs:element name="RequestProcessingState">
1602         <xs:complexType>
1603             <xs:complexContent>
1604                 <xs:restriction base="mows-xs:RequestProcessingStateType"/>
1605             </xs:complexContent>
1606         </xs:complexType>
1607     </xs:element>
1608     <xs:element name="RequestCompletedState">
1609         <xs:complexType>
1610             <xs:complexContent>
1611                 <xs:restriction base="mows-xs:RequestProcessingStateType"/>
1612             </xs:complexContent>
1613         </xs:complexType>
1614     </xs:element>
1615     <xs:element name="RequestFailedState">
1616         <xs:complexType>
1617             <xs:complexContent>
1618                 <xs:restriction base="mows-xs:RequestProcessingStateType"/>
1619             </xs:complexContent>
1620         </xs:complexType>
1621     </xs:element>
1622     <xs:complexType name="MessageContentNotIncludedFlag"/>
1623     <xs:simpleType name="MessageSizeUnitType">
1624         <xs:restriction base="xs:string">
1625             <xs:enumeration value="bit"/>
1626             <xs:enumeration value="byte"/>

```

```

1627         <xs:enumeration value="word"/>
1628         <xs:enumeration value="dword"/>
1629         <xs:enumeration value="qword"/>
1630     </xs:restriction>
1631 </xs:simpleType>
1632 <xs:complexType name="MessageContentSizeType">
1633     <xs:simpleContent>
1634         <xs:extension base="xs:positiveInteger">
1635             <xs:attribute name="Unit"
1636                 type="mows-xs:MessageSizeUnitType" use="required"/>
1637             <xs:anyAttribute namespace="##other" processContents="lax"/>
1638         </xs:extension>
1639     </xs:simpleContent>
1640 </xs:complexType>
1641 <xs:complexType name="MessageContentType">
1642     <xs:sequence>
1643         <xs:element name="Size"
1644             type="mows-xs:MessageContentSizeType" minOccurs="0"/>
1645         <xs:choice>
1646             <xs:element name="NotIncluded"
1647                 type="mows-xs:MessageContentNotIncludedFlag"/>
1648             <xs:element name="Text" type="xs:string"/>
1649             <xs:element name="Binary" type="xs:base64Binary"/>
1650             <xs:element name="Xml"
1651                 type="mows-xs:AnyXmlContentsType"/>
1652         </xs:choice>
1653         <xs:any namespace="##other" processContents="lax"
1654             minOccurs="0" maxOccurs="unbounded"/>
1655     </xs:sequence>
1656     <xs:anyAttribute namespace="##other" processContents="lax"/>
1657 </xs:complexType>
1658 <xs:complexType name="AnyXmlContentsType">
1659     <xs:sequence>
1660         <xs:any namespace="##any" processContents="lax"
1661             minOccurs="0" maxOccurs="unbounded"/>
1662     </xs:sequence>
1663     <xs:anyAttribute namespace="##any" processContents="lax"/>
1664 </xs:complexType>
1665 <xs:complexType name="MessageInformationType">
1666     <xs:sequence>
1667         <xs:element name="TransportInformation"
1668             type="mows-xs:AnyXmlContentsType" minOccurs="0"/>
1669         <xs:element name="Message" type="mows-xs:MessageContentType"/>
1670         <xs:any namespace="##any" processContents="lax"
1671             minOccurs="0" maxOccurs="unbounded"/>
1672     </xs:sequence>
1673     <xs:anyAttribute namespace="##any" processContents="lax"/>
1674 </xs:complexType>
1675 <xs:complexType name="RequestProcessingStateInformationType">
1676     <xs:sequence>
1677         <xs:element ref="muws-xs2:StateTransition" maxOccurs="unbounded"/>
1678     </xs:sequence>
1679 </xs:complexType>
1680 <xs:element name="RequestProcessingNotification">
1681     <xs:complexType>
1682         <xs:sequence>

```

```

1683         <xs:element name="Request"
1684             type="mows-xs:MessageInformationType"
1685             minOccurs="0"/>
1686         <xs:element name="Reply"
1687             type="mows-xs:MessageInformationType"
1688             minOccurs="0"/>
1689         <xs:element name="StateInformation"
1690             type="mows-
1691 xs:RequestProcessingStateInformationType"/>
1692         <xs:any namespace="##any" processContents="lax"
1693             minOccurs="0" maxOccurs="unbounded"/>
1694     </xs:sequence>
1695     <xs:attribute name="CurrentTime" type="xs:dateTime" use="required"/>
1696     <xs:anyAttribute namespace="##any" processContents="lax"/>
1697 </xs:complexType>
1698 </xs:element>
1699 <xs:simpleType name="IPv4AddressType">
1700     <xs:restriction base="xs:hexBinary">
1701         <xs:length value="8" fixed="true"/>
1702     </xs:restriction>
1703 </xs:simpleType>
1704 <xs:element name="IPv4Address" type="mows-xs:IPv4AddressType"/>
1705 <xs:simpleType name="IPv6AddressType">
1706     <xs:restriction base="xs:hexBinary">
1707         <xs:length value="32" fixed="true"/>
1708     </xs:restriction>
1709 </xs:simpleType>
1710 <xs:element name="IPv6Address" type="mows-xs:IPv6AddressType"/>
1711 <xs:simpleType name="TcplpDirectionType">
1712     <xs:restriction base="xs:string">
1713         <xs:enumeration value="to"/>
1714         <xs:enumeration value="from"/>
1715     </xs:restriction>
1716 </xs:simpleType>
1717 <xs:simpleType name="TcplpProtocolType">
1718     <xs:restriction base="xs:string">
1719         <xs:enumeration value="TCP"/>
1720         <xs:enumeration value="UDP"/>
1721     </xs:restriction>
1722 </xs:simpleType>
1723 <xs:element name="TcplpInfo">
1724     <xs:complexType>
1725         <xs:sequence>
1726             <xs:choice>
1727                 <xs:element ref="mows-xs:IPv4Address"/>
1728                 <xs:element ref="mows-xs:IPv6Address"/>
1729             </xs:choice>
1730             <xs:any namespace="##any" processContents="lax"
1731                 minOccurs="0" maxOccurs="unbounded"/>
1732         </xs:sequence>
1733         <xs:attribute name="Direction" type="mows-xs:TcplpDirectionType"
1734             use="required"/>
1735         <xs:attribute name="Port" type="xs:positiveInteger" use="required"/>
1736         <xs:attribute name="Protocol" type="mows-xs:TcplpProtocolType"
1737             use="required"/>
1738         <xs:anyAttribute namespace="##any" processContents="lax"/>

```

```
1739         </xs:complexType>
1740     </xs:element>
1741 </xs:schema>
1742
```

Appendix E. WSDL elements

```

1744 <?xml version="1.0" encoding="utf-8"?>
1745 <definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
1746 xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:wsrf-rp="http://docs.oasis-
1747 open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.xsd" xmlns:mows-
1748 xs="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.xsd" xmlns:mows-
1749 wsdl="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.wsdl"
1750 targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.wsdl">
1751
1752     <types>
1753         <xs:schema elementFormDefault="qualified"
1754 targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.wsdl">
1755             <xs:import namespace="http://docs.oasis-
1756 open.org/wsdm/2004/12/mows/wsdm-mows.xsd" schemaLocation="http://docs.oasis-
1757 open.org/wsdm/2004/12/mows/wsdm-mows.xsd"/>
1758             </xs:schema>
1759         </types>
1760
1761         <message name="GetManageabilityReferencesRequest">
1762             <part name="body" element="mows-xs:GetManageabilityReferences"/>
1763         </message>
1764         <message name="GetManageabilityReferencesResponse">
1765             <part name="body"
1766                 element="mows-xs:GetManageabilityReferencesResponse"/>
1767         </message>
1768
1769         <portType name="ManageabilityReferences">
1770             <operation name="GetManageabilityReferences">
1771                 <input name="GetManageabilityReferencesRequest"
1772                     message="mows-wsdl:GetManageabilityReferencesRequest"/>
1773                 <output name="GetManageabilityReferencesResponse"
1774                     message="mows-wsdl:GetManageabilityReferencesResponse"/>
1775             </operation>
1776         </portType>
1777
1778         <portType name="EndpointIdentification"
1779             wsrf-rp:ResourceProperties="mows-xs:EndpointIdentificationProperties"/>
1780
1781         <portType name="EndpointMetrics"
1782             wsrf-rp:ResourceProperties="mows-xs:EndpointMetricsProperties"/>
1783
1784         <portType name="EndpointOperationalState"
1785             wsrf-rp:ResourceProperties="mows-xs:EndpointOperationalStateProperties"/>
1786     </definitions>
1787

```

Appendix F. Notification topic spaces

```

1789 <?xml version="1.0" encoding="utf-8"?>
1790 <wstop:TopicSpace name="MOWS"
1791     targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows-
1792 events.xml"
1793     xmlns:muws-xs2="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1794 part2.xsd"
1795     xmlns:muws-xs1="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1796 part1.xsd"
1797     xmlns:mows-xs="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.xsd"
1798     xmlns:wstop="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.xsd">
1799
1800 <wstop:Topic name="IdentificationCapability" messageTypes="muws-xs1:ManagementEvent"/>
1801 <wstop:Topic name="MetricsCapability" messageTypes="muws-xs1:ManagementEvent"/>
1802 <wstop:Topic name="OperationalStateCapability" messageTypes="muws-
1803 xs1:ManagementEvent"/>
1804 <wstop:Topic name="OperationalStatusCapability" messageTypes="muws-
1805 xs1:ManagementEvent"/>
1806 <wstop:Topic name="RequestProcessingStateCapability" messageTypes="muws-
1807 xs1:ManagementEvent"/>
1808
1809
1810 <wstop:Topic name="RequestProcessingObservations"
1811     messageTypes="muws-xs1:ManagementEvent">
1812     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1813 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1814 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1815 xs:RequestProcessingNotification)=1]
1816     </wstop:MessagePattern>
1817     <wstop:Topic name="RequestReceived"
1818         messageTypes="muws-xs1:ManagementEvent">
1819         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1820 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1821 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1822 xs:RequestProcessingNotification)=1]
1823         </wstop:MessagePattern>
1824     </wstop:Topic>
1825     <wstop:Topic name="RequestProcessing"
1826         messageTypes="muws-xs1:ManagementEvent">
1827         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1828 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1829 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1830 xs:RequestProcessingNotification)=1]
1831         </wstop:MessagePattern>
1832     </wstop:Topic>
1833     <wstop:Topic name="RequestCompleted"
1834         messageTypes="muws-xs1:ManagementEvent">
1835         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1836 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1837 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1838 xs:RequestProcessingNotification)=1]
1839         </wstop:MessagePattern>

```



```

1840     </wstop:Topic>
1841     <wstop:Topic name="RequestFailed"
1842         messageTypes="muws-xs1:ManagementEvent">
1843         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1844 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1845 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1846 xs:RequestProcessingNotification)=1]
1847         </wstop:MessagePattern>
1848     </wstop:Topic>
1849     <wstop:Topic name="Digest"
1850         messageTypes="muws-xs1:ManagementEvent">
1851         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1852 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1853 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1854 xs:RequestProcessingNotification)=1]
1855         </wstop:MessagePattern>
1856     </wstop:Topic>
1857 </wstop:Topic>
1858
1859 <wstop:Topic name="RequestProcessingObservationsWithAttachments"
1860     messageTypes="muws-xs1:ManagementEvent">
1861     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1862 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1863 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1864 xs:RequestProcessingNotification)=1]
1865     </wstop:MessagePattern>
1866     <wstop:Topic name="RequestReceived"
1867         messageTypes="muws-xs1:ManagementEvent">
1868         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1869 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1870 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1871 xs:RequestProcessingNotification)=1]
1872         </wstop:MessagePattern>
1873     </wstop:Topic>
1874     <wstop:Topic name="RequestProcessing"
1875         messageTypes="muws-xs1:ManagementEvent">
1876         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1877 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1878 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1879 xs:RequestProcessingNotification)=1]
1880         </wstop:MessagePattern>
1881     </wstop:Topic>
1882     <wstop:Topic name="RequestCompleted"
1883         messageTypes="muws-xs1:ManagementEvent">
1884         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1885 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1886 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1887 xs:RequestProcessingNotification)=1]
1888         </wstop:MessagePattern>
1889     </wstop:Topic>
1890     <wstop:Topic name="RequestFailed"
1891         messageTypes="muws-xs1:ManagementEvent">
1892         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1893 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1894 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1895 xs:RequestProcessingNotification)=1]

```

```
1896     </wstop:MessagePattern>
1897     </wstop:Topic>
1898     <wstop:Topic name="Digest"
1899         messageTypes="muws-xs1:ManagementEvent">
1900         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1901 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1902 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1903 xs:RequestProcessingNotification)=1]
1904         </wstop:MessagePattern>
1905     </wstop:Topic>
1906 </wstop:Topic>
1907 </wstop:TopicSpace>
```