



Web Services Distributed Management: Management of Web Services (WSDM-MOWS) 1.1

OASIS-Committee Specification, June 2, 2006

Document identifier:

wsdm-mows-1.1-spec-cs-01

Location:

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

Editors:

Kirk Wilson, Computer Associates kirk.wilson@ca.com

Igor Sedukhin, Computer Associates.

Abstract:

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.

Status:

This document is an OASIS Committee Specification.

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.

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/>).

This specification makes direct and indirect normative references to evolving specifications in OASIS and W3C. The WSDM TC intends to move to referencing the standard specification as they become available.

The errata document for this specification is maintained at:

<http://docs.oasis-open.org/wsdm/wsdm-mows-1.1-errata.pdf>

35 Table of Contents

36	1	Introduction	4
37	1.1	Terminology.....	4
38	1.2	Notational conventions.....	5
39	2	Architecture.....	6
40	2.1	In-band and Out-of-band Manageability	7
41	2.2	Application to Resources Exposed as Web Services	7
42	2.3	Self-Management.....	8
43	3	Managing Web Services	9
44	3.1	Responsibilities of the Implementations of the Manageability Endpoints	9
45	3.2	Manageability at the Web service level.....	10
46	3.3	Using manageability of Web services endpoints	10
47	4	Security Considerations	12
48	4.1	Additional security considerations when managing Web services	12
49	5	Web service manageability capabilities	14
50	5.1	Common manageability capabilities.....	14
51	5.1.1	Manageability References	15
52	5.1.1.1	Operations.....	16
53	5.1.1.1.1	GetManageabilityReferences.....	16
54	5.2	Web service endpoint manageability capabilities	16
55	5.2.1	Identity	16
56	5.2.2	Identification.....	17
57	5.2.2.1	Properties	17
58	5.2.2.2	Events.....	18
59	5.2.3	Metrics	18
60	5.2.3.1	Information markup declarations	19
61	5.2.3.2	Properties	19
62	5.2.3.3	Events.....	22
63	5.2.4	Operation Metrics.....	22
64	5.2.4.1	Properties	23
65	5.2.4.2	Events.....	24
66	5.2.5	Operational State	24
67	5.2.5.1	Information markup declarations	24
68	5.2.5.2	Properties	26
69	5.2.5.3	Events.....	26
70	5.2.6	Operational Status	27
71	5.2.6.1	Events.....	27
72	5.2.7	Operation Operational Status	28
73	5.2.7.1	Properties	28
74	5.2.7.2	Events.....	29
75	5.2.8	Request Processing State	29
76	5.2.8.1	Information markup declarations	30
77	5.2.8.2	Events.....	31

78	5.2.8.2.1	RequestProcessingNotification message.....	35
79	5.2.8.2.2	Examples of events against the Web service endpoint request processing state...	37
80	6	References.....	40
81	6.1	Normative	40
82	6.2	Non-normative.....	40
83		Appendix A. Acknowledgments	42
84		Appendix B. Revision History	43
85		Appendix C. Notices	45
86		Appendix D. XML Schemas.....	46
87		Appendix E. WSDL elements	54
88		Appendix F. Notification topic spaces.....	55
89			

90 1 Introduction

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

99

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

103

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

- 106 ▪ WS Architecture **[WS-Arch]**
- 107 ▪ XML **[XML]**
- 108 ▪ XML Namespaces **[XNS]**
- 109 ▪ XML Schema **[XMLS]**
- 110 ▪ SOAP **[SOAP]**
- 111 ▪ WSDL **[WSDL]**
- 112 ▪ WS-Addressing **[WS-A]**
- 113 ▪ WS-ResourceProperties **[WS-RP]**
- 114 ▪ WS-BaseNotification **[WS-N]**
- 115 ▪ WS-Topics **[WS-T]**
- 116 ▪ XML Path Language **[XPath]**

117

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

120

121 1.1 Terminology

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

125

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

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

129

130 1.2 Notational conventions

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

- 134 ▪ The syntax appears as an XML instance, but the values indicate the data types instead of
135 values.
- 136 ▪ {any} is a placeholder for elements from some other namespace (like ##other in XML
137 Schema).
- 138 ▪ Characters are appended to attributes, elements, and {any} to indicate the number of
139 times they may occur as follows: ? (0 or 1), * (0 or more), + (1 or more). No character
140 indicates exactly 1 occurrence. The characters [and] are used to indicate that contained
141 items are to be treated as a group with respect to the ?, *, and + characters.
- 142 ▪ Attributes, elements, and values separated by | and grouped with (and) are meant to be
143 syntactic alternatives.
- 144 ▪ ... is used in XML start elements to indicate that attributes from some other namespace
145 are allowed.
- 146 ▪ The XML namespace prefixes are used to indicate the namespace of the element being
147 defined

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

150

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

- 154 ▪ Path = '/'? (['@' ? (NCName | QName | '*')] | ['(' (NCName | QName | '*') ') '] '/' Path) ?
- 155 ▪ NCName is an XML non-qualified name as defined by XML Schema [XMLS]. In this case
156 the namespace is assumed to default to the namespace of this specification.
- 157 ▪ QName is an XML qualified name as defined by XML Schema [XMLS].
- 158 ▪ The symbol * denotes any name match.
- 159 ▪ The symbol / denotes a path delimiter. If it appears as the first element of the path, it
160 denotes the root of the XML document.
- 161 ▪ The symbol @ denotes a reference to an XML attribute, otherwise NCName, QName or *
162 refer to an XML element.
- 163 ▪ The symbols (and) denote a reference to an XML Schema type.

164

165 For example, /E1/E2/@A1 refers to an attribute A1 of an element E2 contained in element E1
166 which is a root of the XML document. E1/ns1:E2/E3 refers to an element E3 which is contained in
167 the element E2 which is contained in the element E1 anywhere in the XML document. In this case
168 element E2 belongs to the namespace mapped to the prefix ns1. (ns2:T1)/E1/ns1:E2/@A1 refers
169 to an attribute A1 on an element E2 contained in the element E1 declared in the XML Schema
170 type T1 which target namespace is mapped to the prefix ns2.

171

172

2 Architecture

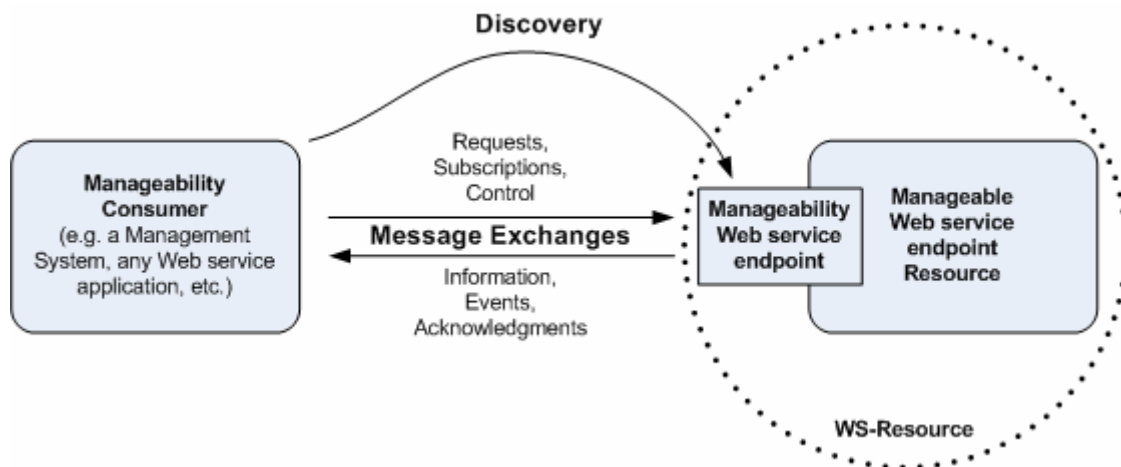
173

174 Management of Web services (MOWS) is an application of Management using Web services
175 (MUWS) to the resources that are elements of the Web Services Architecture [WS-Arch]. This
176 WSDM specification defines how the manageability of Web service endpoints and resources
177 exposed as Web services can be accessed via Web services. In order to achieve this goal,
178 MOWS is based on the MUWS specifications, and the architecture, definitions and dependencies
179 thereof [MUWS].

180

181 Application of the WSDM architecture concepts (section 2 of the MUWS specification part 1) to
182 the management of Web services could be described as follows (Figure 1). A *manageability Web*
183 *service endpoint* (or, shortly, *manageability endpoint*) provides access to the *manageable Web*
184 *service endpoint resource* (a manageable resource, in terms of MUWS). A manageable Web
185 service endpoint (or, shortly, *manageable endpoint*) could be, for example, an endpoint of an
186 order entry Web service for which received messages could be counted and reported to the
187 *manageability consumers*. Following the WSDM concepts, the manageability consumer discovers
188 the manageability endpoint and exchanges messages with it in order to request information,
189 subscribe to events or control the manageable endpoint resource.

190



191

192

Figure 1. Management of Web services concepts

193

194 Refer to section 2 of the MUWS specification part 1 [MUWS] for more detailed explanation of
195 discovery and message exchange between manageability consumers and manageability
196 endpoints.

197

198 The following are important aspects of the WSDM architecture.. Please refer to the referenced
199 sections of the MUWS specification [MUWS]

200

- **Focus on resources** (section 2.1 of MUWS part 1) – focus on providing access to the manageable resources – a contract between a manageability consumer and a manageable resource with regards to discovery and message exchanges.

201

202

203

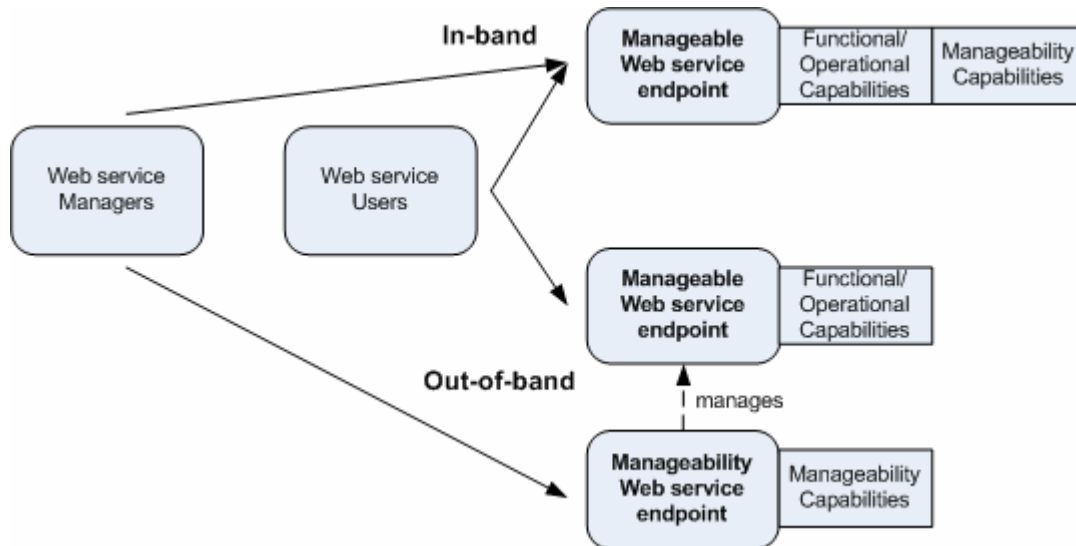
- **Composeability** (section 2.2 of MUWS part 1) – allows a non-conflicting, incremental mix of Web services implementation aspects and manageability capabilities.

204

205
206
207
208
209
210
211
212
213

2.1 In-band and Out-of-band Manageability

A unique feature 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.



214
215
216

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

217

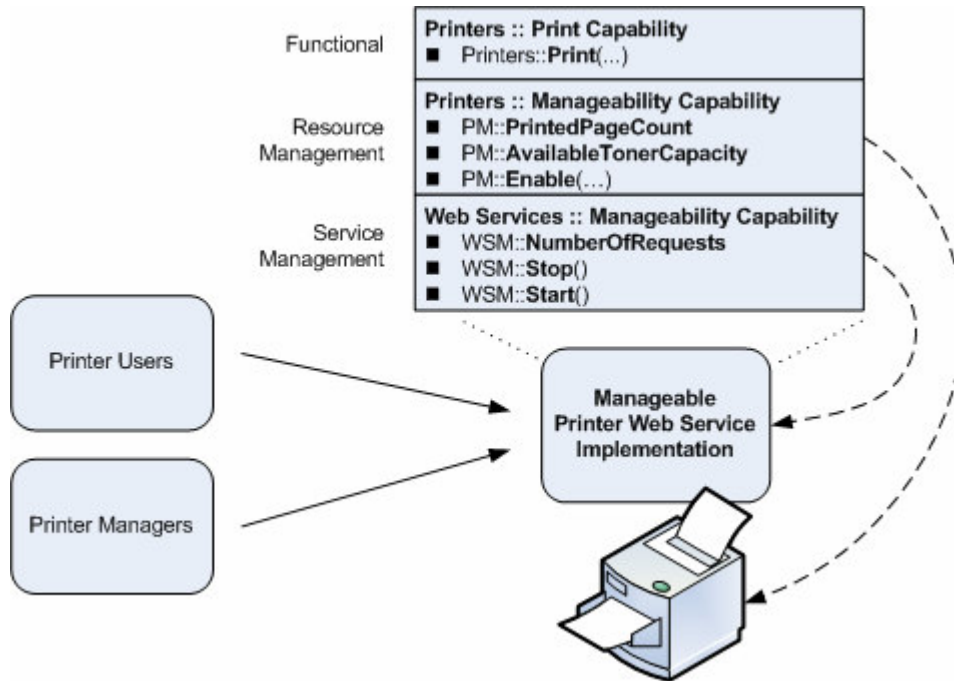
2.2 Application to Resources Exposed as Web Services

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

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

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

232

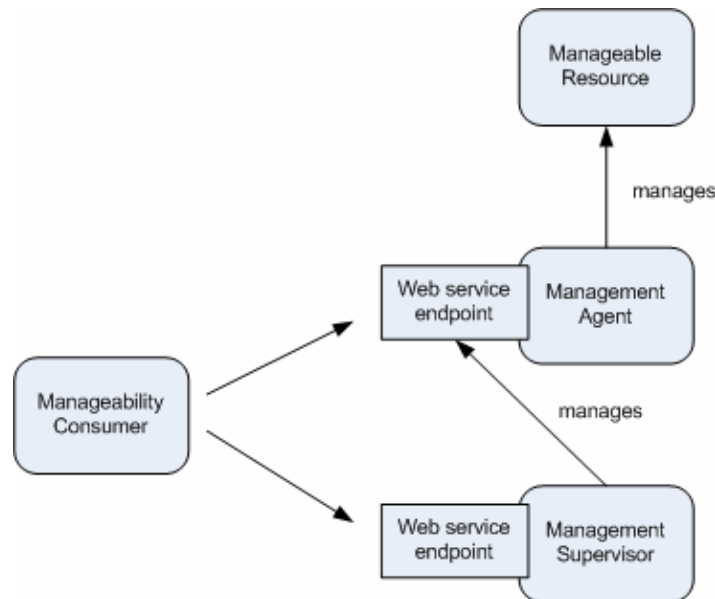


233
234

Figure 3. Application to resources exposed as Web services

235 **2.3 Self-Management**

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



241
242
243

Figure 4. Applying MOWS to MUWS

244 3 Managing Web Services

245 Using definitions expressed in WSDL 1.1 [WSDL] and WS-Addressing [WS-A] as guidelines, a
246 Web service (described by a WSDL 1.1 service element) is an aggregate of endpoints (described
247 by WSDL 1.1 port elements). An endpoint binds a Web service interface (described by a WSDL
248 1.1 portType element) to an address (URI). Each interface describes a set of messages that
249 could be exchanged and their format. Properly formatted messages could be sent to the endpoint
250 at the address in the way prescribed by the binding (described by a WSDL 1.1 binding element).
251 A Web service description contains definitions of a combination of interfaces and services.

252

253 According to the section 2, management of Web services starts at an endpoint resource which,
254 therefore, becomes a manageable resource, specifically called a manageable endpoint. The
255 reason the Web service endpoint is the basic manageable resource is that (1) anything behind an
256 endpoint is a concrete implementation (e.g. an application hosted on a server), and (2) an
257 aggregate of endpoints is a logical construct, management of which has to be inferred from
258 manageability of the constituent endpoints. This specification focuses on defining manageability
259 capabilities of the Web service endpoints. Furthermore, (1) is in the realm of the
260 applications/systems/networks management, and (2) should be done by the intelligent
261 management systems. Aspects of (1) are further discussed in section 3.1. Aspects of (2) are
262 further discussed in section 3.2.

263

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

266 3.1 Responsibilities of the Implementations of the Manageability 267 Endpoints

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

273

274 Consider two examples. The first case is that of a hardware routing configuration. A hardware
275 device controls access to all messages sent to a particular URL such as
276 <http://external.example.com/theService>. Upon receipt of messages for that URL, the device
277 distributes the messages to Web service endpoints at the <http://s1.example.com/theService>,
278 <http://s1.example.com/theService>, and <http://s2.example.com/theService> addresses.

279

280 If, say, a query regarding metrics were made regarding the Web service endpoint receiving
281 messages at the <http://external.example.com/theService> address, it is the responsibility of the
282 implementation of the manageability endpoint to aggregate the results from the three underlying
283 Web service endpoints to provide a meaningful response.

284

285 A second example is one in which a single Web service endpoint is accessible at two distinct
286 URLs due to DNS aliasing. Consider the Web service endpoint at
287 <http://services.example.com/creditCheck>. External to the Example Company, this Web service
288 endpoint is accessible at the <http://ourservices.example.com/creditCheck> address, while
289 internally, this Web service endpoint is accessible at <http://extservices.example.com/creditCheck>.

290 However, in both cases, the message processing is performed by the same machine, application,
291 code, etc. The Web service endpoint implementation itself is *aware* of the means by which it is
292 addressed (e.g. is using the URL header of the HTTP messages), and it adjusts message
293 processing appropriately.

294

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

298 **3.2 Manageability at the Web service level**

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

306

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

312 **3.3 Using manageability of Web services endpoints**

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

- 315 1. Obtain an EPR to the manageability endpoint. One of the following ways may be used.
 - 316 a. Discover manageable resources as described in the MUWS specifications
317 **[MUWS]**.
 - 318 b. Exercise the Manageability References capability (section 5.1.1) on the
319 functional Web services endpoint.
 - 320 c. The functional Web services endpoint may also be the manageability endpoint
321 (section 2.1). Determine that by detecting if the endpoint supports the MUWS
322 Identity capability:
 - 323 i. Either, obtain the WSDL document describing the manageability
324 endpoint and look for a ResourceId element (see MUWS specification
325 part 1 section 5.1) in the first level children of the resource properties
326 document root **[WS-RP]**.
 - 327 ii. Or, request the value of the ManageabilityCapability property (see
328 MUWS specification part 1 section 5.2) and look for the URI which
329 identifies the MUWS Identity capability.
- 330 2. Using the EPR obtained in the previous step, and based on the manageability capabilities
331 intended to be used, build Web services messages targeted at the manageable Web
332 services endpoint.
 - 333 a. Obtain the WSDL document describing the manageability endpoint and
334 understand how operations defined by the manageability capabilities are bound.

335
336
337

b. Request the value of the ManageabilityCapability property (see MUWS specification part 1 section 5.2) and look for the URIs which identify the capabilities to be used.

338
339
340
341

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.

342 4 Security Considerations

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

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

- 350 ▪ **Message alteration** – Alteration is prevented by including signatures of the message
351 information using WS-Security.
- 352 ▪ **Message disclosure** – Confidentiality is preserved by encrypting sensitive data using
353 WS-Security.
- 354 ▪ **Key integrity** – Key integrity is maintained by using the strongest algorithms possible.
- 355 ▪ **Authentication** – Authentication is established using the mechanisms described in WS-
356 Security and other related specifications. Each message is authenticated using the
357 mechanisms described in WS-Security.
- 358 ▪ **Accountability** – Accountability is a function of the type of and strength of the key and
359 algorithms being used. In many cases, a strong symmetric key provides sufficient
360 accountability. However, in some environments, strong PKI signatures are required.
- 361 ▪ **Availability** – All services are subject to a variety of availability attacks. Replay detection
362 is a common attack and it is RECOMMENDED that this be addressed by the
363 mechanisms described in WS-Security. Other attacks, such as network-level denial of
364 service attacks are harder to avoid and are outside the scope of this specification. That
365 said, care should be taken to ensure that minimal state is saved prior to any
366 authenticating sequences.

367

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

373

374 4.1 Additional security considerations when managing Web 375 services

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

- 378 ▪ If a manageable Web services endpoint supports messages from both a consumer of a
379 service and a manager of a service section 2.1, it may be important to identify a security
380 model which allows for the appropriate level of granularity with regard to the message
381 origin. For example, setting configuration options may be allowed by a manageability
382 consumer but not an application consumer. When these composed services are
383 deployed, it will be important to understand the authorization model for both management
384 and functional use.
- 385 ▪ In order to make the management systems secure in addition to reliable and accountable
386 (section 2.3), it will be important to follow a set of guidelines and best practices that detail

387 how to compose MOWS with existing security implementations and emerging
388 specifications for authorization and trust.

389 ▪ Implementers of this specification may need to give a particular attention to security when
390 implementing the following manageability capabilities.

391 ○ Manageability References (section 5.1.1) – this capability allows access to the
392 manageability endpoint references of a functional Web service endpoint. The
393 concern is that visibility to these references may need to be protected differently
394 than visibility of the functional Web service endpoint and its operations.

395 ○ Request Processing State (section 5.2.8) – this capability allows managers to
396 subscribe to notifications against request processing by a functional Web service
397 endpoint.

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

401 2. The request messages may be encrypted and signed. Therefore, managers
402 may need to possess information that allows them to deal with such
403 encrypted and signed messages.

404 3. Notification messages which contain information about request messages
405 SHOULD be encrypted to avoid spoofing of this information by intercepting
406 notification messages.

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

410

5 Web service manageability capabilities

411

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

414

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

418

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

423

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

426

Prefix	Namespace
muws1	http://docs.oasis-open.org/wsdm/muws1-2.xsd
muws2	http://docs.oasis-open.org/wsdm/muws2-2.xsd
mows	http://docs.oasis-open.org/wsdm/mows-2.xsd
mowsw	http://docs.oasis-open.org/wsdm/mows-2.wsdl
mowse	http://docs.oasis-open.org/wsdm/mowse-2.xml
wsa	http://www.w3.org/2005/08/addressing
wsdl	http://www.w3.org/2002/07/wsdl
S	http://schemas.xmlsoap.org/soap/envelope/ or http://www.w3.org/2002/12/soap-envelope
xs	http://www.w3.org/2001/XMLSchema
wsrf-rp	http://docs.oasis-open.org/wsrf/rp-2
wstop	http://docs.oasis-open.org/wsn/t-1

427

428 Unless otherwise specified, XML elements and XML schema types introduced in this specification
429 belong to the namespace mapped to the **mows** prefix.

430

5.1 Common manageability capabilities

431

432 The following sections define manageability capabilities applicable to Web services and
433 resources exposed as Web services.

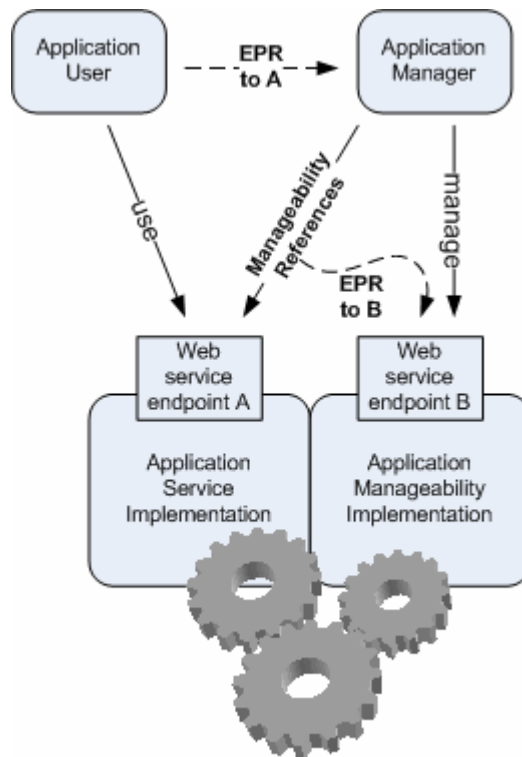
434 **5.1.1 Manageability References**

435 This capability is identified by the following URI:
436 <http://docs.oasis-open.org/mows2/capabilities/ManageabilityReferences>

437
438 This capability allows a functional/operational Web service or a resource exposed as a Web
439 service (section 2.2) (*the service*) to provide references to its manageability endpoints. This
440 capability is intended for implementations of functional/operational Web services endpoints. The
441 consumer may exchange messages with *the service* in order to request references to the
442 manageability endpoints. Using obtained references, the consumer may exchange messages
443 with the manageability endpoints in order to perform management activities to *the service*.

444
445 For example (Figure 5), an application user accesses a Web service endpoint A. The application
446 user then gives the endpoint A reference to the application manager which accesses the Web
447 service endpoint A in order to obtain a reference to the application manageability implementation
448 accessible at the Web service endpoint B. The application manager may now manage the
449 application by exchanging management related messages with endpoint B.

450



451

452 **Figure 5.** Use of Manageability References capability

453

454 The name of this capability identifies its semantics of providing references to manageability
455 endpoints of the service that supports this capability. Within this specification, this capability
456 consists of one operation:

- 457 • GetManageabilityReferences.

458 but applications may associate further operations with this capability.

459 **5.1.1.1 Operations**

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

461

462 **5.1.1.1.1 GetManageabilityReferences**

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

465

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

467

```
468 <GetManageabilityReferences/>
```

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

471 The wsa:Action MUST contain the URI

472 <http://docs.oasis-open.org/wsdm/mows/GetManageabilityReferences>.

473

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

476

```
477 <GetManageabilityReferencesResponse>  
478 <muws1:ManageabilityEndpointReference>  
479 <!-- see [MUWS] -->  
480 </muws1:ManageabilityEndpointReference>+  
481 </GetManageabilityReferencesResponse>
```

482 The wsa:Action MUST contain the URI

483 <http://docs.oasis-open.org/wsdm/mows/GetManageabilityReferencesResponse>.

484

485 **GetManageabilityReferencesResponse** is a GED which identifies the response to the
486 requested GetManageabilityReferences operation.

487

488 **GetManageabilityReferencesResponse/muws1:ManageabilityEndpointReference** is a
489 reference to the Web service endpoint which provides access to the management of the
490 functional/operational Web service endpoint or the Web service-enabled resource which
491 responded to the GetManageabilityReferences operation request message.

492

493 **5.2 Web service endpoint manageability capabilities**

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

495 **5.2.1 Identity**

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

499 5.2.2 Identification

500 This capability is identified by the following URI:

501 **http://docs.oasis-open.org/mows-2/capabilities/Identification**

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

- 503 ▪ `<muws2:Capability>http://docs.oasis-open.org/mows-`
504 `2/capabilities/Identification</muws2:Capability>`

505

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

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

512

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

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

517

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

521 This capability consists of two properties:

- 522 • EndpointReference (mandatory)
523 • EndpointDescriptions (optional)
524 • Description (zero to many)

525 and defines one event: IdentificationCapability,

526 but applications may associate further properties and events with this capability.

527 5.2.2.1 Properties

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

530

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

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

533

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

- 538 ▪ Is not *Mutable*
539 ▪ Is not *Modifiable*

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

- 543 ▪ Is *Mutable*
- 544 ▪ Is not *Modifiable*

545 **5.2.2.2 Events**

546 The following specification defines this capability notification topics in the namespace mapped to
547 the **mowse** prefix.

548

```
549   <wstop:Topic name="IdentificationCapability" messageTypes="muws1:ManagementEvent"/>
```

550

551 **mowse:IdentificationCapability** is a topic on which management events related to this
552 manageability capability SHOULD be emitted.

553

554 Property change events MUST be wrapped in Management events and published on topics
555 defined in this section.

556

557 **5.2.3 Metrics**

558 This capability is identified by the following URI:

559 **<http://docs.oasis-open.org/mows-2/capabilities/Metrics>**

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

- 561 ▪ <muws2:Capability>[http://docs.oasis-open.org/mows-](http://docs.oasis-open.org/mows-2/capabilities/Metrics)
562 2/capabilities/Metrics</muws2:Capability>

563

564 The name of this capability identifies its semantics of providing properties that are useful in
565 measuring the use and performance of Web services. This capability consists of ten properties:

- 566 • NumberOfRequests (optional)
- 567 • NumberOfFailedRequests (optional)
- 568 • NumberOfSuccessfulRequests (optional)
- 569 • ServiceTime (optional)
- 570 • MaxResponseTime (optional)
- 571 • LastResponseTime (optional)
- 572 • MaxRequestSize (optional)
- 573 • LastRequestSize (optional)
- 574 • MaxResponseSize (optional)
- 575 • LastResponseSize (optional)

576 and defines one event: MetricsCapability,

577 but applications may associate further properties and events with this capability.

578

579 This capability extends the definition of the MUWS Metrics capability. WSDM manageable
580 endpoints that intend to support the MOWS **Metrics** capability MUST support the MUWS **Metrics**
581 capability (section 3.4 of the **[MUWS]** part 2) as well.

582

583 It is recommended that for adequate calculations, the Web service endpoint metric properties
584 (one or all) are retrieved together with the **muws2:CurrentTime** property (e.g., using one request
585 to retrieve multiple properties).

586

587 Metrics and request processing states are related. The request processing state change
588 boundaries are the points where metric counters are incremented. These states are defined
589 below, in section 5.2.8.

590 **5.2.3.1 Information markup declarations**

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

593

```
594 <xs:complexType name="IntegerCounter">  
595   <xs:simpleContent>  
596     <xs:extension base="xs:nonNegativeInteger">  
597       <xs:attributeGroup ref="muws2:MetricAttributes"/>  
598       <xs:anyAttribute namespace="##other" processContents="lax"/>  
599     </xs:extension>  
600   </xs:simpleContent>  
601 </xs:complexType>
```

602

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

604

```
605 <xs:complexType name="DurationMetric">  
606   <xs:simpleContent>  
607     <xs:extension base="xs:duration">  
608       <xs:attributeGroup ref="muws2:MetricAttributes"/>  
609       <xs:anyAttribute namespace="##other" processContents="lax"/>  
610     </xs:extension>  
611   </xs:simpleContent>  
612 </xs:complexType>
```

613

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

615 **5.2.3.2 Properties**

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

618

```
619 <NumberOfRequests>IntegerCounter</NumberOfRequests>?  
620 <NumberOfFailedRequests>IntegerCounter</NumberOfFailedRequests>?  
621 <NumberOfSuccessfulRequests>IntegerCounter</NumberOfSuccessfulRequests>?  
622 <ServiceTime>DurationMetric</ServiceTime>?  
623 <MaxResponseTime>DurationMetric</MaxResponseTime>?  
624 <LastResponseTime>DurationMetric</LastResponseTime>?  
625 <MaxRequestSize>IntegerCounter</MaxRequestSize>?  
626 <LastRequestSize>IntegerCounter</LastRequestSize>?  
627 <MaxResponseSize>IntegerCounter</MaxResponseSize>?  
628 <LastResponseSize>IntegerCounter</LastResponseSize>?
```

629

630 **NumberOfRequests** is a counter of the number of request messages that the Web service
631 endpoint has received. This counter is incremented by 1 whenever a request reaches the
632 Received state according to the Request Processing Model (Figure 6). Metadata about this
633 property is as follows.

- 634 ▪ Is *Mutable*
- 635 ▪ Is not *Modifiable*
- 636 ▪ <muws2:ChangeType>Counter</muws2:ChangeType>
- 637 ▪ <muws2:TimeScope>SinceReset</muws2:TimeScope> or
- 638 <muws2:TimeScope>Interval</muws2:TimeScope>
- 639 ▪ <muws2:GatheringTime>OnChange</muws2:GatheringTime>

640 **NumberOfFailedRequests** is a counter of the number of request messages that the Web service
641 endpoint has received, and a (SOAP) fault was sent in reply. This counter is incremented by 1
642 whenever a request reaches the Failed state according to the Request Processing Model (Figure
643 6). Metadata about this property is as follows.

- 644 ▪ Is *Mutable*
- 645 ▪ Is not *Modifiable*
- 646 ▪ <muws2:ChangeType>Counter</muws2:ChangeType>
- 647 ▪ <muws2:TimeScope>SinceReset</muws2:TimeScope> or
- 648 <muws2:TimeScope>Interval</muws2:TimeScope>
- 649 ▪ <muws2:GatheringTime>OnChange</muws2:GatheringTime>

650 **NumberOfSuccessfulRequests** is a counter of the number of request messages that the Web
651 service endpoint has received, and anything but a (SOAP) fault was sent in reply. This counter is
652 incremented by 1 whenever a request reaches the Completed state according to the Request
653 Processing Model (Figure 6). Metadata about this property is as follows.

- 654 ▪ Is *Mutable*
- 655 ▪ Is not *Modifiable*
- 656 ▪ <muws2:ChangeType>Counter</muws2:ChangeType>
- 657 ▪ <muws2:TimeScope>SinceReset</muws2:TimeScope> or
- 658 <muws2:TimeScope>Interval</muws2:TimeScope>
- 659 ▪ <muws2:GatheringTime>OnChange</muws2:GatheringTime>

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

662

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

- 665 ▪ Is *Mutable*
- 666 ▪ Is not *Modifiable*
- 667 ▪ <muws2:ChangeType>Counter</muws2:ChangeType>
- 668 ▪ <muws2:TimeScope>SinceReset</muws2:TimeScope> or
- 669 <muws2:TimeScope>Interval</muws2:TimeScope>
- 670 ▪ <muws2:GatheringTime>OnChange</muws2:GatheringTime>

671 **MaxResponseTime** is a gauge indicating the maximum time duration between all requests
672 received and their completion or failure. Metadata about this property is as follows.

- 673 ▪ Is *Mutable*
- 674 ▪ Is not *Modifiable*

- 675 ▪ <muws2:ChangeType>Gauge</muws2:ChangeType>
- 676 ▪ <muws2:TimeScope>SinceReset</muws2:TimeScope> or
- 677 ▪ <muws2:TimeScope>Interval</muws2:TimeScope>
- 678 ▪ <muws2:GatheringTime>OnChange</muws2:GatheringTime>

679 **LastResponseTime** is a gauge indicating the last recorded time duration between the last
 680 request received and its completion or failure. Metadata about this property is as follows.

- 681 ▪ Is *Mutable*
- 682 ▪ Is not *Modifiable*
- 683 ▪ <muws2:ChangeType>Gauge</muws2:ChangeType>
- 684 ▪ <muws2:TimeScope>PointInTime</muws2:TimeScope>
- 685 ▪ <muws2:GatheringTime>OnChange</muws2:GatheringTime>

686 **MaxRequestSize** is a gauge indicating the maximum size in bytes for all requests received
 687 regardless of their completion or failure. Metadata about this property is as follows.

- 688 ▪ Is *Mutable*
- 689 ▪ Is not *Modifiable*
- 690 ▪ <muws2:Units>byte</muws:Units>
- 691 ▪ <muws2:ChangeType>Gauge</muws2:ChangeType>
- 692 ▪ <muws2:TimeScope>SinceReset</muws2:TimeScope> or
- 693 ▪ <muws2:TimeScope>Interval</muws2:TimeScope>
- 694 ▪ <muws2:GatheringTime>OnChange</muws2:GatheringTime>

695 **LastRequestSize** is a gauge indicating the last request size in bytes for the last request received
 696 regardless of the completion or failure of the request. Metadata about this property is as follows.

- 697 ▪ Is *Mutable*
- 698 ▪ Is not *Modifiable*
- 699 ▪ <muws2:Units>byte</muws:Units>
- 700 ▪ <muws2:ChangeType>Gauge</muws2:ChangeType>
- 701 ▪ <muws2:TimeScope>PointInTime</muws2:TimeScope>
- 702 ▪ <muws2:GatheringTime>OnChange</muws2:GatheringTime>

703 **MaxResponseSize** is a gauge indicating the maximum response size in bytes for all responses
 704 sent regardless of their completion or failure. Metadata about this property is as follows.

- 705 ▪ Is *Mutable*
- 706 ▪ Is not *Modifiable*
- 707 ▪ <muws2:Units>byte</muws:Units>
- 708 ▪ <muws2:ChangeType>Gauge</muws2:ChangeType>
- 709 ▪ <muws2:TimeScope>SinceReset</muws2:TimeScope> or
- 710 ▪ <muws2:TimeScope>Interval</muws2:TimeScope>
- 711 ▪ <muws2:GatheringTime>OnChange</muws2:GatheringTime>

712 **LastResponseSize** is a gauge indicating the size of the last response in bytes sent regardless of
 713 the completion or failure of the request. Metadata about this property is as follows.

- 714 ▪ Is *Mutable*
- 715 ▪ Is not *Modifiable*
- 716 ▪ <muws2:Units>byte</muws:Units>
- 717 ▪ <muws2:ChangeType>Gauge</muws2:ChangeType>

- 718 ▪ <muws2:TimeScope>PointInTime</muws2:TimeScope>
- 719 ▪ <muws2:GatheringTime>OnChange</muws2:GatheringTime>

720

721 Note that if a metric property has a <muws2:TimeScope>SinceReset</muws2:TimeScope>
722 metadata value, the muws2:ResetAt attribute MUST be reported on the property element and the
723 muws2:Duration attribute MUST NOT be reported. If a metric property has a
724 <muws2:TimeScope>Interval</muws2:TimeScope> metadata value, the muws2:ResetAt attribute
725 MAY be reported on the property element and the muws2:Duration attribute MUST be reported.

726

727 Also note that in this specification, counters are not just monotonically increasing variables, but
728 also represent a cumulative metric of some kind e.g. number of requests over time. Gauges, on
729 the other hand, do not represent a cumulative metric, but rather discrete values of some kind at a
730 particular time (e.g. response time).

731 5.2.3.3 Events

732 The following specification defines this capability notification topics in the namespace mapped to
733 the **mowse** prefix.

734

```
735 <wstop:Topic name="MetricsCapability" messageTypes="muws1:ManagementEvent"/>
```

736

737 **mowse:MetricsCapability** is a topic on which management events related to this manageability
738 capability SHOULD be emitted.

739

740 Property change events MUST be wrapped in Management events and published on topics
741 defined in this section.

742

743 5.2.4 Operation Metrics

744 This capability is identified by the following URI:

745 **<http://docs.oasis-open.org/wsdm/mows-2/capabilities/OperationMetrics>**

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

- 747 ▪ <muws2:Capability>[http://docs.oasis-open.org/wsdm/mows-](http://docs.oasis-open.org/wsdm/mows-2/capabilities/OperationMetrics)
748 2/capabilities/OperationMetrics</muws2:Capability>

749

750 The name of this capability identifies its semantics of providing MOWS metrics at an operational
751 level. This capability applies MOWS metrics to operations within the managed Web service. It
752 consists of multiply occurring complex properties, each complex property represents an operation
753 of the service that may be characterized with MOWS metrics (see section 5.2.3.2):

- 754 • OperationMetrics (many)
 - 755 • NumberOfRequests (optional)
 - 756 • NumberOfFailedRequests (optional)
 - 757 • NumberOfSuccessfulRequests (optional)
 - 758 • ServiceTime (optional)
 - 759 • MaxResponseTime (optional)
 - 760 • LastResponseTime (optional)

- 761 • MaxRequestSize (optional)
- 762 • LastRequestSize (optional)
- 763 • MaxResponseSize (optional)
- 764 • LastResponseSize (optional)

765 and defines one event: OperationMetricsCapability,
 766 but applications may associate further properties and events with this capability.
 767

768 WSDM manageable endpoints that intend to support the MOWS **Operation Metrics** capability
 769 MUST support the MUWS **Metrics** capability (section 3.4 of the [MUWS] part 2) as well. It is
 770 expected that support of the **Operation Metrics** capability will usually be done in addition to the
 771 MOWS **Metrics** capability, but this is not a requirement.

772

773 It is recommended that for adequate calculations, the Web service operation metric properties
 774 (one or all) are retrieved together with the **muws2:CurrentTime** property.

775

776 5.2.4.1 Properties

777 The following is the specification of the Web service endpoint operation metrics complex property
 778 (i.e. XML element that represents the metrics of an operation).

779

```

780 <OperationMetrics operationName="xs:NCName", portType="xs:QName"? ...>*
781   <NumberOfRequests>IntegerCounter</NumberOfRequests>?
782   <NumberOfFailedRequests>IntegerCounter</NumberOfFailedRequests>?
783   <NumberOfSuccessfulRequests>IntegerCounter</NumberOfSuccessfulRequests>?
784   <ServiceTime>DurationMetric</ServiceTime>?
785   <MaxResponseTime>DurationMetric</MaxResponseTime>?
786   <LastResponseTime>DurationMetric</LastResponseTime> ?
787   <MaxRequestSize>IntegerCounter</MaxRequestSize>?
788   <LastRequestSize>IntegerCounter</LastRequestSize>?
789   <MaxResponseSize>IntegerCounter</MaxResponseSize>?
790   <LastResponseSize>IntegerCounter</LastResponseSize>?
791   {any} *
792 </OperationMetrics>
  
```

793

794 **OperationMetrics** is the group property that contains the metrics for one operation within the
 795 managed Web service. The operation to which the metrics apply is identified by the attributes of
 796 this element.

797 **OperationMetrics/@operationName** is the name of operation to which the metrics apply. This is
 798 a required attribute. The value of this attribute is the name of the operation as it is specified in the
 799 portType of the Web service

800 **OperationMetrics/@portType** is the QName of the portType in which the operation occurs. This
 801 is an optional attribute. The value of this attribute is the QName of the portType as it is specified
 802 in the WSDL of the Web service.

803 The metrics contained within the OperationMetrics complex property are the MOWS metrics
 804 defined in section 5.2.3.2.

805

806 **5.2.4.2 Events**

807 The following specification defines the capability notification topics in the namespace mapped to
808 the **mowse** prefix.

809

```
810 <wstop:Topic name="OperationMetricsCapability" messageTypes="muws1:ManagementEvent"/>
```

811

812 **mowse:OperationMetricsCapability** is a topic on which management events related to this
813 manageability capability SHOULD be emitted.

814

815 **5.2.5 Operational State**

816 This capability is identified by the following URI:

817 **<http://docs.oasis-open.org/mows-2/capabilities/OperationalState>**

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

- 819 ▪ `<muws2:Capability>http://docs.oasis-open.org/mows-`
820 `2/capabilities/OperationalState</muws2:Capability>`

821

822 The name of this capability identifies its semantics of providing information on the operational
823 state of a managed Web service. This capability consists of two properties:

- 824 • `CurrentOperationalState` (mandatory)
- 825 • `LastOperationStateTransition` (zero to many)

826 and defines one event: `OperationalStateCapability`,

827 but applications may associate further properties and events with this capability.

828

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

833

834 **5.2.5.1 Information markup declarations**

835 Each state MUST be identified by a QName and represented by a corresponding XML element.

836 Following is a list of elements corresponding to the operational states of the Web service

837 endpoint operation according to the WSLC state model [**WSLC**].

- 838 ▪ **UpState**

839 This element corresponds to the WSLC UP top-level state which means that the Web
840 service endpoint operation is capable of accepting new requests. This state is the parent
841 state of the BUSY and IDLE substates, as defined below.

- 842 ▪ **DownState**

843 This element corresponds to the WSLC DOWN top-level state which means that the Web
844 service endpoint operation is not capable of accepting new requests. This state is the
845 parent state of the STOPPED, CRASHED, and SATURATED substates, as defined
846 below.

- 847 ▪ **BusyState**
- 848 This element corresponds to the WSLC BUSY substate of UP which means that the Web
- 849 service endpoint operation is capable of accepting new requests during processing of
- 850 other requests. This element MUST contain the UpState element.
- 851 ▪ **IdleState**
- 852 This element corresponds to the WSLC IDLE substate of UP which means that the Web
- 853 service endpoint operation is capable of accepting new requests and is not processing
- 854 any other requests. This element MUST contain the UpState element.
- 855 ▪ **StoppedState**
- 856 This element corresponds to the WSLC STOPPED substate of DOWN which means that
- 857 the Web service endpoint operation is not capable of accepting new requests and was
- 858 intentionally stopped by an administrator. This element MUST contain the DownState
- 859 element.
- 860 ▪ **CrashedState**
- 861 This element corresponds to the WSLC CRASHED substate of DOWN which means that
- 862 the Web service endpoint operation is not capable of accepting new requests as a result
- 863 of some internal failure. This element MUST contain the DownState element
- 864 ▪ **SaturatedState**
- 865 This element corresponds to the WSLC SATURATED substate of DOWN which means
- 866 that the Web service endpoint operation is not capable of accepting new requests due to
- 867 lack of resources. This element MUST contain the DownState element.

868

869 It is possible to extend the above state model. Substates MAY be introduced and MUST be

870 identified by QNames, however, new top-level operational states MUST NOT be defined. In order

871 to represent the taxonomy lineage of substates in XML, the MUWS approach is used (section 3.2

872 in the **[MUWS]** part 2).

873

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

875

```
876 <xs:complexType name="OperationalStateType">
877 <xs:complexContent>
878     <xs:extension base="muws2:StateType"/>
879 </xs:complexContent>
880 </xs:complexType>
```

881

882 The **OperationalStateType** is used to declare elements which contain any valid elements

883 designating an operational state of a Web service endpoint.

884

- 885 ▪ A substate of the operational state MUST be declared according to the following rules.
- 886 ○ An XML element is declared with a QName which identifies the desired substate
- 887 semantics, for example my-app:DatabaseCleanupState
- 888 ○ The contents of the XML element MUST be the only element which corresponds
- 889 to the generalized state, for example mows:StoppedState

890

891 An instance of the request processing state information represented in XML may look as shown

892 in the following example,

893

```
894 <my:OperationalStateInformationElement xsi:type="mows:OperationalStateType">
895     <my-app:DatabaseCleanupState>
896         <mows:StoppedState>
897             <mows:DownState/>
898         </mows:StoppedState>
899     </my-app:DatabaseCleanupState>
900 </my:RequestProcessingStateInformationElement>
```

901

902 5.2.5.2 Properties

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

905

```
906 <CurrentOperationalState>mows:OperationalStateType</CurrentOperationalState>
907 <LastOperationalStateTransition>
908     muws2:StateTransitionType
909 </LastOperationalStateTransition> ?
```

910

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

- 913 ▪ Is *Mutable*
- 914 ▪ Is not *Modifiable*

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

- 918 ▪ Is *Mutable*
- 919 ▪ Is not *Modifiable*

920

921 5.2.5.3 Events

922 The following specification defines this capability notification topics in the namespace mapped to
923 the **mwse** prefix.

924

```
925 <wstop:Topic name="OperationalStateCapability" messageTypes="muws1:ManagementEvent"/>
```

926

927 **mwse:OperationalStateCapability** is a topic on which management events related to this
928 manageability capability SHOULD be emitted.

929

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

934

935 5.2.6 Operational Status

936 This capability is identified by the following URI:

937 **http://docs.oasis-open.org/mows-2/capabilities/OperationalStatus**

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

- 939 ▪ `<muws2:Capability>http://docs.oasis-open.org/mows-`
940 `2/capabilities/OperationalStatus</muws2:Capability>`

941

942 The name of this capability indicates its semantics of indicating the operational status of a
943 managed Web Service. This capability consists of one property:

- 944 • **OperationalStatus** (mandatory), as defined in the MUWS part 2 **OperationalStatus** capability
945 and defines one event: **OperationalStatusCapability**,
946 but applications may associate further properties and events with this capability.

947

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

952

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

- 955 ▪ The WSLC **UP** state **MUST** be reported as the **Available** contents of the
956 **muws2:OperationalStatus** property. Any sub-state of WSLC **UP** **MUST** be reported as
957 **Available**.
- 958 ▪ The WSLC **DOWN** state **MUST** be reported as the **Unavailable** contents of the
959 **muws2:OperationalStatus** property. Any sub-state of WSLC **DOWN** **SHOULD** be
960 reported as **Unavailable**. The **STOPPED** and **CRASHED** substates of WSLC **DOWN**
961 **MUST** be reported as **Unavailable**.
- 962 ▪ The WSLC **SATURATED** sub-state of **DOWN** **MAY** be reported as the
963 **PartiallyAvailable** contents of the **muws2:OperationalStatus** property.

964

965 5.2.6.1 Events

966 The following specification defines this capability notification topics in the namespace mapped to
967 the **mowse** prefix.

968

```
969 <wstop:Topic name="OperationalStatusCapability"  
970 messageTypes="muws1:ManagementEvent"/>
```

971

972 **mowse:OperationalStatusCapability** is a topic on which management events related to this
973 manageability capability **SHOULD** be emitted.

974

975 Property change events **MUST** be wrapped in Management events and published on topics
976 defined in this section.

977

978 5.2.7 Operation Operational Status

979 This capability is identified by the following URI:

980 <http://docs.oasis-open.org/wsdm/mows-2/capabilities/OperationOperationalStatus>

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

- 982 ▪ `<muws2:Capability>http://docs.oasis-open.org/wsdm/mows-`
983 `2/capabilities/OperationOperationalStatus</muws2:Capability>`

984 An operation operational status property reflects whether a named operation is available,
985 unavailable, or degraded. Operation operational status does not conform to a specific state
986 model. Operation operational status is related to the Request Processing Model of the [WSLC],
987 see Figure 6, but does not directly map to request processing states. The manageable resource
988 provides the appropriate mapping from request states to the status of an operation and sets the
989 *OperationOperationalStatus* property accordingly (see below).

990 The name of this capability identifies its semantics indicating the operational status of an
991 operation within a managed Web service. For each operation, there is one complex property that
992 provides the *OperationStatus* of that operation:

- 993 • *OperationOperationalStatus* (many)
994 • *OperationStatus*, as defined in section 5.2.6

995 and the capability defines one event: *OperationOperationalStatusCapability*

996 but applications may associate further properties and events with this capability.

997

998 5.2.7.1 Properties

999 The following is the specification of the Web service endpoint operation operational status
1000 complex property (i.e. XML element that represents the operational status of an operation).

```
1001 <OperationOperationalStatus operationName="xs:NCName", portType="xs:QName"? ...>*  
1002   <muws2:OperationalStatus>  
1003     (Available|PartiallyAvailable|Unavailable|Unknown)  
1004   </muws2:OperationalStatus>  
1005   {any} *  
1006 </OperationOperationalStatus>
```

1007 **OperationOperationalStatus** is the complex property representing each operation for which
1008 operational status information can be provided.

1009 **OperationOperationalStatus/@operationName** is the name of operation to which the
1010 operational status applies. This is a required attribute. The value of this attribute is the name of
1011 the operation as it is specified in the portType of the Web service

1012 **OperationOperationalStatus/@portType** is the QName of the portType in which the operation
1013 occurs. This is an optional attribute. The value of this attribute is the name of the portType as it is
1014 specified in the WSDL of the Web service.

1015 **OperationOperationalStatus/muws2:OperationalStatus** is the operational status property that
1016 is defined in MUWS Part 2 and used as well in the MOWS *OperationStatus* capability; see
1017 section 5.2.6. The valid values of this property with their intended interpretation in this context
1018 are:

- 1019 • *Available*: This value indicates that the named operation is operating normally within any
1020 configured operating parameters, and is able to perform its function. It is capable or
1021 receiving requests and completing processing.
1022 • *PartiallyAvailable*: This value indicates that the named operation is operating, but outside
1023 of configured operating parameters. An operation reporting this operation operational
1024 status is able to complete some requests but may fail others. For example, an operation

1025 may be failing only for a subset of requests because of database table required by only
1026 that subset is locked.
1027 • *Unavailable*: This value indicates that the named operation is not operating, and is not
1028 able to perform any functional tasks. The operation may be unable to received requests
1029 and it may be failing all requests.
1030 • *Unknown*: This value indicates that the named operation is unable to report status at this
1031 time.
1032

1033 5.2.7.2 Events

1034 The following specification defines this capability notification topics in the namespace mapped to
1035 the **mowse** prefix.

1036

```
1037 <wstop:Topic name="OperationOperationalStatusCapability"  
1038 messageTypes="muws1:ManagementEvent"/>
```

1039

1040 **mowse:OperationOperationalStatusCapability** is a topic on which management events related
1041 to this manageability capability SHOULD be emitted.

1042

1043 Property change events MUST be wrapped in Management events and published on topics
1044 defined in this section.

1045

1046 5.2.8 Request Processing State

1047 This capability is identified by the following URI:

1048 **<http://docs.oasis-open.org/mows-2/capabilities/RequestProcessingState>**

1049 All properties, operations and events defined for this capability have the following metadata:

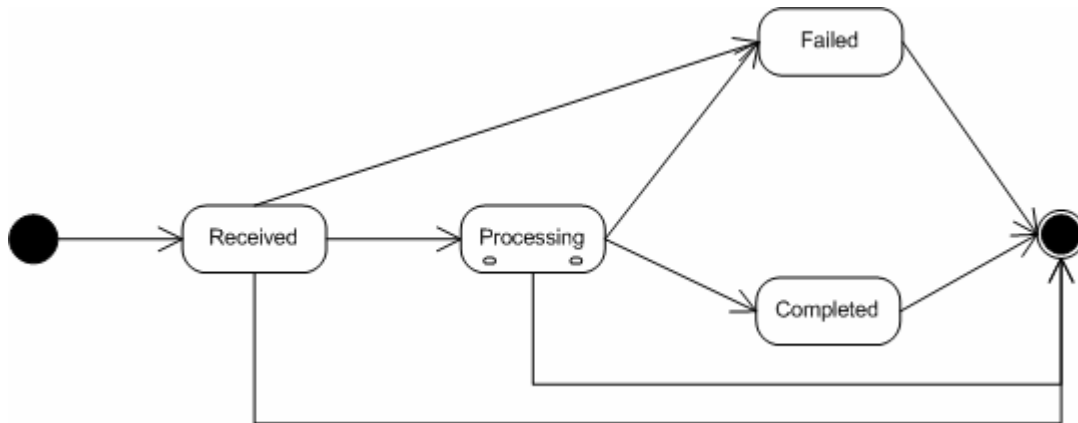
1050 • <muws2:Capability>[http://docs.oasis-open.org/mows-](http://docs.oasis-open.org/mows-2/capabilities/RequestProcessingState)
1051 [2/capabilities/RequestProcessingState](http://docs.oasis-open.org/mows-2/capabilities/RequestProcessingState)</muws2:Capability>

1052

1053 The name of this capability identifies its semantics of providing information on the processing
1054 state of a request to a managed Web service. This capability consists of no properties but defines
1055 12 events:

- 1056 • RequestProcessingObservations
- 1057 • RequestProcessingObservations/RequestReceived
- 1058 • RequestProcessingObservations/RequestProcessing
- 1059 • RequestProcessingObservations/RequestComplete
- 1060 • RequestProcessingObservations/RequestFailed
- 1061 • RequestProcessingObservations/Digest
- 1062 • RequestProcessingObservationsWithAttachments
- 1063 • RequestProcessingObservationsWithAttachments/RequestReceived
- 1064 • RequestProcessingObservationsWithAttachments/RequestProcessing
- 1065 • RequestProcessingObservationsWithAttachments/RequestComplete
- 1066 • RequestProcessingObservationsWithAttachments/RequestFailed

1067 • RequestProcessingObservationsWithAttachment/Digest
 1068 but applications may associate further properties and events with this capability.
 1069 By the definition, a Web service endpoint accepts and processes messages targeted at it –
 1070 *requests*. Every request goes through a number of states (e.g. received, processing, completed
 1071 or failed) as defined by the [WSLC] and extended here.
 1072



1073
 1074 **Figure 6.** Request processing states
 1075

1076 The state diagram represents a model in which states MAY have duration and transitions are
 1077 instantaneous. When extending this model one MUST extend only the Processing compound
 1078 state.
 1079

1080 **5.2.8.1 Information markup declarations**

1081 Each state MUST be identified by a QName and represented by a corresponding XML element.
 1082 Following is a list of elements corresponding to the top-level states of the request processing
 1083 state model (Figure 6).

- 1084 ▪ **RequestReceivedState**
 1085 This element corresponds to the Received top-level state which means that the Web
 1086 service endpoint has accepted a request to perform one of the service's functional
 1087 responsibilities. This state represents the earliest point at which the manageability
 1088 provider knows that the request was dispatched to the Web service endpoint being
 1089 managed.
- 1090 ▪ **RequestProcessingState**
 1091 This element corresponds to the Processing top-level state which means that the Web
 1092 service endpoint is doing some internal processing/execution to fulfill the requested
 1093 function. This state represents the earliest point at which the application module or
 1094 business logic begins processing the request. For example, if the application server
 1095 queues the request before dispatching it to the business logic, the time difference
 1096 between "request received" and "processing" will include the duration the request was
 1097 queued.
- 1098 ▪ **RequestCompletedState**
 1099 This element corresponds to the Completed top-level state which means that the Web
 1100 service endpoint successfully completed requested function returning results to the
 1101 requester.

1102

- **RequestFailedState**

1103

This element corresponds to the Failed top-level state which means that the Web service endpoint encountered an error and didn't complete the requested function, returning error/fault to the requester.

1104

1105

1106

1107

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

1108

1109

1110

1111

1112

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

1113

1114

```
<xs:complexType name="RequestProcessingStateType">
```

1115

```
<xs:complexContent>
```

1116

```
  <xs:extension base="muws2:StateType"/>
```

1117

```
</xs:complexContent>
```

1118

```
</xs:complexType>
```

1119

1120

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

1121

1122

1123

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

1124

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

1125

1126

The contents of the XML element MUST be the only element which corresponds to the generalized state, for example muws2:RequestProcessingState

1127

1128

1129

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

1130

1131

1132

```
<my:RequestProcessingStateInformationElement
```

1133

```
  xsi:type="mows:RequestProcessingStateType">
```

1134

```
    <my-soap:SerializationState>
```

1135

```
      <mows:RequestProcessingState/>
```

1136

```
    </my-soap:SerializationState>
```

1137

```
</my:RequestProcessingStateInformationElement>
```

1138

1139

5.2.8.2 Events

1140

Notifications are emitted when requests enter one of the request processing states (Figure 6).

1141

1142

Property change events MUST be wrapped in Management events and published on topics defined in this section.

1143

1144

1145

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

1146

1147

1148

```
1149 <wstop:Topic name="RequestProcessingStateCapability"
1150 messageTypes="muws1:ManagementEvent"/>
1151
1152 <wstop:Topic name="RequestProcessingObservations"
1153   messageTypes="muws1:ManagementEvent">
1154   <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1155   //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
1156   and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
1157   </wstop:MessagePattern>
1158   <wstop:Topic name="RequestReceived"
1159     messageTypes="muws1:ManagementEvent">
1160     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1161     //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
1162     and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
1163     </wstop:MessagePattern>
1164     </wstop:Topic>
1165     <wstop:Topic name="RequestProcessing"
1166       messageTypes="muws1:ManagementEvent">
1167       <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1168       //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
1169       and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
1170       </wstop:MessagePattern>
1171       </wstop:Topic>
1172       <wstop:Topic name="RequestCompleted"
1173         messageTypes="muws1:ManagementEvent">
1174         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1175         //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
1176         and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
1177         </wstop:MessagePattern>
1178         </wstop:Topic>
1179         <wstop:Topic name="RequestFailed"
1180           messageTypes="muws1:ManagementEvent">
1181           <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1182           //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
1183           and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
1184           </wstop:MessagePattern>
1185           </wstop:Topic>
1186           <wstop:Topic name="Digest"
1187             messageTypes="muws1:ManagementEvent">
1188             <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1189             //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
1190             and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
1191             </wstop:MessagePattern>
1192             </wstop:Topic>
1193           </wstop:Topic>
1194         </wstop:Topic>
1195       </wstop:Topic name="RequestProcessingObservationsWithAttachments"
1196         messageTypes="muws1:ManagementEvent">
1197         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1198         //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
1199         and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
1200         </wstop:MessagePattern>
1201         <wstop:Topic name="RequestReceived"
1202           messageTypes="muws1:ManagementEvent">
```

```

1203     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1204 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
1205 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
1206     </wstop:MessagePattern>
1207     </wstop:Topic>
1208     <wstop:Topic name="RequestProcessing"
1209         messageTypes="muws1:ManagementEvent">
1210     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1211 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
1212 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
1213     </wstop:MessagePattern>
1214     </wstop:Topic>
1215     <wstop:Topic name="RequestCompleted"
1216         messageTypes="muws1:ManagementEvent">
1217     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1218 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
1219 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
1220     </wstop:MessagePattern>
1221     </wstop:Topic>
1222     <wstop:Topic name="RequestFailed"
1223         messageTypes="muws1:ManagementEvent">
1224     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1225 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
1226 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
1227     </wstop:MessagePattern>
1228     </wstop:Topic>
1229     <wstop:Topic name="Digest"
1230         messageTypes="muws1:ManagementEvent">
1231     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1232 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
1233 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
1234     </wstop:MessagePattern>
1235     </wstop:Topic>
1236 </wstop:Topic>

```

1237

1238 **mowse:RequestProcessingStateCapability** is a topic on which management events related to
1239 this manageability capability SHOULD be emitted.

1240 **mowse:ManageableEndpoint/mowse:RequestProcessingObservations** indicates availability
1241 of any information about the processing of any request by the Web service endpoint (Figure 6) as
1242 observed by the implementation of a manageable Web service.

1243 The notification message for this topic MUST contain at most one
1244 **RequestProcessingNotification** element (defined in section 5.2.8.2.1). The MUWS
1245 management event MUST also declare the event situation category with the
1246 muws2:ReportSituation element and the severity value "1" (Informational). It is
1247 recommended to subscribe to these topics with proper preconditions and selectors using
1248 expressions against the contents of the RequestProcessingNotification element.

1249 **mowse:ManageableEndpoint/mowse:RequestProcessingObservations/mowse:RequestRe**
1250 **ceived** indicates that a request was received by the Web service endpoint being managed
1251 (Received state in Figure 6). The notification message format for this topic is the same as the
1252 notification message format for the
1253 mowse:ManageableEndpoint/mowse:RequestProcessingObservations topic. This is a state
1254 change event and therefore notification messages MUST contain exactly one

1255 muws2:StateTransition element inside of the RequestProcessingNotification/StateInformation
1256 element.

1257 **mowse:ManageableEndpoint/mowse:RequestProcessingObservations/mowse:RequestPro**
1258 **cessing** indicates that a request is being processed by the Web service endpoint being managed
1259 (Processing state in Figure 6). The notification message format for this topic is the same as the
1260 notification message format for the
1261 mowse:ManageableEndpoint/mowse:RequestProcessingObservations topic. This is a state
1262 change event and therefore notification messages **MUST** contain exactly one
1263 muws2:StateTransition element inside of the RequestProcessingNotification/StateInformation
1264 element.

1265 **mowse:ManageableEndpoint/mowse:RequestProcessingObservations/mowse:RequestCo**
1266 **mpleted** indicates that a request was successfully completed by the Web service endpoint being
1267 managed (Completed state in Figure 6). The notification message format for this topic is the
1268 same as the notification message format for the
1269 mowse:ManageableEndpoint/mowse:RequestProcessingObservations topic. This is a state
1270 change event and therefore notification messages **MUST** contain exactly one
1271 muws2:StateTransition element inside of the RequestProcessingNotification/StateInformation
1272 element.

1273 **mowse:ManageableEndpoint/mowse:RequestProcessingObservations/mowse:RequestFail**
1274 **ed** indicates that a request was failed (not successfully completed) by the Web service endpoint
1275 being managed (Failed state in Figure 6). The notification message format for this topic is the
1276 same as the notification message format for the
1277 mowse:ManageableEndpoint/mowse:RequestProcessingObservations topic. This is a state
1278 change event and therefore notification messages **MUST** contain exactly one
1279 muws2:StateTransition element inside of the RequestProcessingNotification/StateInformation
1280 element.

1281 **mowse:ManageableEndpoint/mowse:RequestProcessingObservations/mowse:Digest**
1282 indicates availability of summary information about a request processed by the Web service
1283 endpoint being managed. The notification message format for this topic is the same as the
1284 notification message format for the
1285 mowse:ManageableEndpoint/mowse:RequestProcessingObservations topic. This is a digest
1286 event and therefore notification messages **MUST** contain one or more muws2:StateTransition
1287 elements inside of the RequestProcessingNotification/StateInformation element. Each
1288 muws2:StateTransition element describes a state transition which occurred with that one request
1289 which this summary notification is informing about. Each state transition information element
1290 carries an attribute indicating the time when that particular transition occurred. Using this
1291 information the manageability consumer can reconstruct the sequence of events with regards to
1292 the request.

1293 **mowse:ManageableEndpoint/mowse:RequestProcessingObservationsWithAttachments**
1294 topic and all of its subtopics are defined exactly as the
1295 mowse:ManageableEndpoint/mowse:RequestProcessingObservations topic and its respective
1296 subtopics, except that the notification messages **MUST** include attachments (if any) of the
1297 request and reply messages sent to/from the Web service endpoint being managed.

1298 The notification message format for this topic and all of its subtopics is the same as the
1299 notification message format for the
1300 mowse:ManageableEndpoint/mowse:RequestProcessingObservations topic, except that
1301 attachments may be sent along with the message. The precise mechanism of sending
1302 the attachment is dependent on 1) the binding of the notification consumer Web service
1303 endpoint **[WS-N]** and 2) the binding of the Web service endpoint being managed.

1304

1305 The mowse:ManageableEndpoint/mowse:RequestProcessingObservations/mowse:
1306 RequestProcessing topic MAY be extended with custom subtopics in order to represent custom
1307 request processing substates of the Processing compound state (Figure 6).

1308

1309 Note that the result of the message pattern XPath expressions in the topic declarations above is
1310 the XML nodeset **[XPath]** of the notification messages that are sent inside of the S:Body element
1311 or the wsnt:Notify element **[WS-N]**.

1312

1313 Note that for the XPath expressions defined here the prefix-to-namespace mapping context
1314 MUST include all prefixes which appear in the XPath expression and mapped according to the
1315 table in the section 5.

1316

1317 **5.2.8.2.1 RequestProcessingNotification message**

1318 The RequestProcessingNotification message format is defined as follows.

1319

```
1320 <RequestProcessingNotification CurrentTime="xs:dateTime" ...>  
1321 <Request ...>  
1322   <TransportInformation ...> {any}* </TransportInformation> ?  
1323   <Message ...>  
1324     <Size Unit=("bit" | "byte" | "word" | "dword" | "qword")  
1325       ...>xs:positiveInteger</Size> ?  
1326     (  
1327       <NotIncluded/> |  
1328       <Text>xs:string</Text> |  
1329       <Binary>xs:base64Binary</Binary> |  
1330       <Xml>{any}* </Xml>  
1331     )  
1332     {any}*  
1333   </Message>  
1334   {any}*  
1335 </Request> ?  
1336 <Reply ...>  
1337   <!-- ... see contents of the Request element above ... -->  
1338 </Reply> ?  
1339 <StateInformation>  
1340 <muws2:StateTransition> <!-- ...see [MUWS]... --> </muws2:StateTransition> +  
1341 </StateInformation>  
1342 {any}*  
1343 </RequestProcessingNotification>
```

1344

1345 **RequestProcessingNotification** is a container element of the information about a request going
1346 through the request processing states (Figure 6).

1347 **RequestProcessingNotification/@CurrentTime** indicates current time measured at the
1348 manageability endpoint. All time/date values in this notification information are synchronized with
1349 this time indication.

1350 **RequestProcessingNotification/Request** element contains information about the request itself.
1351 Note that the request is not necessarily serialized as a SOAP message. Therefore, the contents
1352 allow information about requests in general, however the information has to be serializable in
1353 XML **[XML]**. The presence of this element in the notification MUST indicate presence of the

1354 actual request message sent to the Web service endpoint being managed. The contents may
1355 vary depending on what the implementation of the manageability endpoint can or intends to
1356 provide. For example, for security reasons the actual contents of the message may be omitted.
1357 However, in order to indicate that the request message exists, this element has to be included in
1358 the notification.

1359 **RequestProcessingNotification/Request/TransportInformation** element contains information
1360 about the transport by which the request was received. The content of this element is open, but
1361 WSDM defines the following elements useable for TCP/IP transports.

```
1362 <TcpIpInfo  
1363     Direction=("from" | "to")  
1364     Port="xs:positiveInteger"  
1365     Protocol=("TCP" | "UDP") ...>  
1366     (  
1367     <IPV4Address>  
1368         xs:hexBinary[xs:length[@value="8" and @fixed="true"]]  
1369     </IPV4Address> |  
1370     <IPV6Address>  
1371         xs:hexBinary[xs:length[@value="32" and @fixed="true"]]  
1372     </IPV6Address>  
1373     )  
1374     {any}*  
1375 </TcpIpInfo>
```

1376 **TcpIpInfo** contains information about a communication to or from an IP addressable
1377 network device.

1378 **TcpIpInfo/@Direction** indicates communication to or from the IP addressable network
1379 device.

1380 **TcpIpInfo/@Port** is a TCP/IP network port number used on the IP addressable network
1381 device.

1382 **TcpIpInfo/@Protocol** indicates if the TCP or UDP protocol is used.

1383 **TcpIpInfo/IPV4Address** contains hexadecimal representation of the IP address version
1384 4. The value MUST represent 32 bits.

1385 **TcpIpInfo/IPV6Address** contains hexadecimal representation of the IP address version
1386 6. The value MUST represent 128 bits.

1387 **RequestProcessingNotification/Request/Message** element contains the message observed by
1388 the Web service endpoint being managed.

1389 **RequestProcessingNotification/Request/Message/Size** indicates size of the message. When
1390 subscribed to observations with attachments, this value includes the size of the message payload
1391 plus all the attachments. Otherwise, just the payload of the message (i.e. size of the contents of
1392 the RequestProcessingNotification/Request/Message element) is reported. Note that the actual
1393 message contents may not be reported for security reasons, however size may be reported.

1394 **RequestProcessingNotification/Request/Message/Size/@Unit** indicates what units were used
1395 to calculate the size of the message. The valid values of this attribute are:

1396 **bit** – size indicates number of bits in the message.

1397 **byte** – size indicates number of bytes (8 bit sets) in the message

1398 **word** – size indicates number of double bytes (16 bit sets) in the message.

1399 **dword** – size indicates number of double words (32 bit sets) in the message.

1400 **qword** – size indicates number of quad words (64 bit sets) in the message.

1401 **RequestProcessingNotification/Request/Message/NotIncluded** element indicates that the
1402 message content is intentionally not provided by the implementation of the Web service endpoint
1403 manageability.

1404 **RequestProcessingNotification/Request/Message/Text** element contains the observed
1405 message's text representation. For example, a non-well formed XML message should be
1406 represented as text. It is recommended that text data is wrapped in an XML CDATA section
1407 **[XML]**.

1408 **RequestProcessingNotification/Request/Message/Binary** element contains the binary
1409 representation of the observed message. If a message cannot be represented as either well-
1410 formed XML nor as text, it should be binary encoded.

1411 **RequestProcessingNotification/Request/Message/Xml** element contains the observed
1412 message's XML representation. For example, a SOAP message envelope element (S:Envelope)
1413 may appear in the contents.

1414 **RequestProcessingNotification/Request/{any}** is an extensibility element where additional
1415 information about the request MAY appear. The form of the information representation in XML is
1416 manageability endpoint implementation specific. In other words, vendor extensions may appear
1417 here.

1418 The **RequestProcessingNotification/Reply** element contains information about the reply (if any)
1419 for the request. Note that fault is also a valid reply element. The content of this element has the
1420 same format as the RequestProcessingNotification/Request element.

1421 **RequestProcessingNotification/StateInformation** element contains information about the
1422 request processing state.

1423 **RequestProcessingNotification/StateInformation/muws2:StateTransition** element contains
1424 information about a state transition. There MUST be exactly one such element for each state
1425 change event. There MUST be one or more such elements for the digest event.

1426 **RequestProcessingNotification/StateInformation/muws2:StateTransition/@muws2:Time**
1427 indicates time when the described transition occurred. Note that according to the request
1428 processing state model (Figure 6), all transitions are instantaneous. Time is measured at the
1429 implementation of the manageability endpoint and is synchronized with the
1430 RequestProcessingNotification/@CurrentTime value reading.

1431 **RequestProcessingNotification/StateInformation/muws2:StateTransition/muws2:EnteredSt**
1432 **ate** indicates which request processing state was entered.

1433 **RequestProcessingNotification/StateInformation/muws2:StateTransition/muws2:Previous**
1434 **State** indicates which request processing state was exited.

1435 **RequestProcessingNotification/{any}** is an extensibility element where additional information
1436 about this request processing notification MAY appear. The form of the information representation
1437 in XML is manageability endpoint implementation specific. In other words, vendor extensions may
1438 appear here.

1439 The contents of the RequestProcessingNotification element SHOULD be used to specify
1440 selectors **[WS-N]** when subscribing to notification messages containing this element.

1441 **5.2.8.2.2 Examples of events against the Web service endpoint request** 1442 **processing state**

1443 Consider the following message exchange with a fictitious order-entry Web service endpoint.

1444

1445 Request:

```
1446 <S:Envelope xmlns:x="..." ... >
1447 . . .
1448 <S:Body>
```

```
1449     <x:Order>
1450         <x:Item>...</x:Item>
1451         <x:Quantity>...</x:Quantity>
1452     </x:Order>
1453 </S:Body>
1454 </S:Envelope>
```

1455
1456 Reply:

```
1457 <S:Envelope xmlns:x="..." ... >
1458 . . .
1459 <S:Body>
1460     <x:Shipped>
1461         <x:Item>...</x:Item>
1462         <x:Quantity>...</x:Quantity>
1463     </x:Shipped>
1464 </S:Body>
1465 </S:Envelope>
```

1466
1467 To be notified of a particular item shortage when the order request is processed and the shipped
1468 quantity is less than the ordered quantity, the following XPath selector should be specified when
1469 subscribing to the
1470 **mowse:ManageableEndpoint/mowse:RequestProcessingObservations/mowse:RequestCo**
1471 **mpleted** topic.

1472
1473 Selector:
1474 `boolean(//mows:RequestProcessingNotification[mows:Request/mows:Message/mows:Xml//x:Ord`
1475 `er/x:Quantity < mows:Reply/mows:Message/mows:Xml//x:Shipped/x:Quantity)`

1476
1477 This way, when the condition is met, the manageable Web service endpoint will emit the
1478 notification message containing the **RequestProcessingNotification** element with the following
1479 contents.

```
1480  
1481 <RequestProcessingNotification CurrentTime="...">
1482 <Request>
1483     <TransportInformation>
1484         <TcpIpInfo Direction="from" Port="2840" Protocol="TCP">
1485             <IPV4Address>C0A80002</IPV4Address>
1486         </TcpIpInfo>
1487         <TcpIpInfo Direction="to" Port="80" Protocol="TCP">
1488             <IPV4Address>C0A80003</IPV4Address>
1489         </TcpIpInfo>
1490     </TransportInformation>
1491     <Message>
1492     <Size Unit="byte">257</Size>
1493     <Xml>
1494     <S:Envelope xmlns:S="..." xmlns:x="..." ...>
1495     . . .
1496     <S:Body>
1497         <x:Order>
1498             <x:Item>123</x:Item>
1499             <x:Quantity>10</x:Quantity>
```

```

1500         </x:Order>
1501     </S:Body>
1502 </S:Envelope>
1503 </Xml>
1504 </Message>
1505 </Request>
1506 <Reply>
1507     <TransportInformation>
1508         <TcpIpInfo Direction="to" Port="2840" Protocol="TCP">
1509             <IPv4Address>C0A80002</IPv4Address>
1510         </TcpIpInfo>
1511         <TcpIpInfo Direction="from" Port="80" Protocol="TCP">
1512             <IPv4Address>C0A80003</IPv4Address>
1513         </TcpIpInfo>
1514     </TransportInformation>
1515     <Message>
1516     <Size Unit="byte">232</Size>
1517     <Xml>
1518     <S:Envelope xmlns:S="..." xmlns:x="..." ...>
1519     . . .
1520     <S:Body>
1521         <x:Shipped>
1522             <x:Item>123</x:Item>
1523             <x:Quantity>2</x:Quantity>
1524         </x:Shipped>
1525     </S:Body>
1526 </S:Envelope>
1527 </Xml>
1528 </Message>
1529 </Reply>
1530 <muws2:StateTransition Time="...">
1531 <muws2:EnteredState/><RequestCompletedState/></muws2:EnteredState>
1532 <muws2:PreviousState><RequestProcessingState/></muws2:PreviousState>
1533 </muws2:StateTransition>
1534 . . .
1535 </RequestProcessingNotification>
1536
1537

```

1538

6 References

1539

6.1 Normative

- 1540 [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
1541 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 1542 [MUWS] V. Bullard, et al., *Web Services Distributed Management:Management*
1543 *using Web Services (MUWS 1.1) Part 1*, <http://docs.oasis->
1544 [open.org/wsdm/wsdm-muws1-1.1-cs-01.pdf](http://docs.oasis-open.org/wsdm/wsdm-muws1-1.1-cs-01.pdf)
- 1545 V. Bullard, et al., *Web Services Distributed Management:Management*
1546 *using Web Services (MUWS 1.1) Part 2*, <http://docs.oasis->
1547 [open.org/wsdm/wsdm-muws2-1.1-cs-01.pdf](http://docs.oasis-open.org/wsdm/wsdm-muws2-1.1-cs-01.pdf)
- 1548 [WS-A] Don Box, et al., *Web services Addressing (WS-Addressing)*,
1549 <http://www.w3.org/TR/ws-addr-core>
- 1550 [WS-RP] Steve Graham, et al., *Web Services Resource Properties 1.2 (WS-*
1551 *ResourceProperties)*, <http://docs.oasis-open.org/wsr/wsrf->
1552 [ws_resource_properties-1.2-spec-os-01.pdf](http://docs.oasis-open.org/wsr/wsrf-ws_resource_properties-1.2-spec-os-01.pdf)
- 1553 [WS-N] Steve Graham, et al., *Web Services Base Notification 1.2 (WS-*
1554 *BaseNotification)*, <http://docs.oasis-open.org/wsn/wsn->
1555 [ws_base_notification-1.3-spec-pr-03.pdf](http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-pr-03.pdf)
- 1556 [WS-T] William Vambenepe, *Web Services Topics 1.2 (WS-Topics)*,
1557 http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-pr-02.pdf
- 1558 [WSDL] Erik Christensen, et al., *Web services Description Language (WSDL)*
1559 *1.1*, W3C Note, March 2001, <http://www.w3.org/TR/wSDL>
- 1560 [SOAP] Don Box, et al., *Simple Object Access Protocol 1.1*, W3C Note, May
1561 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- 1562 [XMLS] Henry S. Thompson, et al. *XML Schema Part 1: Structures*, W3C
1563 Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-1/>
1564 Paul V. Biron, et al. *XML Schema Part 2: Datatypes*, W3C
1565 Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-2/>
- 1566 [XML] Tim Bray, et al., *Extensible Markup Language (XML) 1.0 (Third Edition)*,
1567 W3C Recommendation, February 2004, <http://www.w3.org/TR/REC-xml>
- 1568 [XNS] Tim Bray, et al., *Namespaces in XML*, W3C Recommendation, January
1569 1999, <http://www.w3.org/TR/REC-xml-names/>
- 1570 [XPath] James Clark, et al., *XML Path Language (XPath) Version 1.0*, W3C
1571 Recommendation, November 1999, [http://www.w3.org/TR/1999/REC-](http://www.w3.org/TR/1999/REC-xpath-19991116)
1572 [xpath-19991116](http://www.w3.org/TR/1999/REC-xpath-19991116)

1573

6.2 Non-normative

- 1574 [MOWS-Reqs] Mark Potts, et al., *WSDM Management of Web Services Requirements*,
1575 October 2003, <http://www.oasis->
1576 [open.org/apps/org/workgroup/wsdm/download.php/3887/WSDM-MOWS-](http://www.oasis-open.org/apps/org/workgroup/wsdm/download.php/3887/WSDM-MOWS-)
1577 [Requirements.20031008.doc](http://www.oasis-open.org/apps/org/workgroup/wsdm/download.php/3887/WSDM-MOWS-Requirements.20031008.doc)
- 1578 [WS-Arch] David Booth, et al. *Web Services Architecture*, W3C Working Group
1579 Note, February 2004, <http://www.w3.org/TR/2004/NOTE-ws-arch->
1580 [20040211/](http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/)
- 1581 [WSLC] Hao He, et al., *Web Service Management: Service Lifecycle*, W3C Note,
1582 February 2004, <http://www.w3.org/TR/2004/NOTE-wslc-20040211/>

1583	[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
1584		
1585		
1586	[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
1587		
1588		
1589		

Appendix A. Acknowledgments

1590

1591 The following people made contributions to the WSDM MOWS Version 1.1 specification: Brian
1592 Carroll, Fred Carter, John DeCarlo, Andreas Dharmawan, Maryann Hondo, Heather Kreger,
1593 Bryan Murray, Michael Perks, Igor Sedukhin, William Vambenepe, Andrea Westerinen.

1594 The following individuals were members of the committee when the specification 1.0 version was
1595 approved by the technical committee: Guru Bhat, Jeff Bohren, Winston Bumpus, Nick Butler,
1596 Brian Carroll, Fred Carter, Michael Clements, David Cox, John DeCarlo, Andreas Dharmawan,
1597 Mark Ellison, John Fuller, Paul Lipton, Heather Kreger, Hal Lockhart, Frederico Maciel, Tom
1598 Maguire, Bryan Murray, Richard Nikula, Mark Peel, Richard Pelavin, Homayoun Pourheidari,
1599 Warren Roberts, Karl Schopmeyer, Igor Sedukhin, David Snelling, Thomas Studwell, William
1600 Vambenepe, Andrea Westerinen, Jim Willits, Zhili Zhang.

1601

1602 The following people made contributions to the WSDM MOWS Version 1.1 specification: Barry
1603 Atkins, Vaughn Bullard, Fred Carter, David Cox, Mark Ellison, Heather Kreger, Fred Maciel,
1604 Bryan Murray, Kirk Wilson with special thanks to Kirk Wilson and Mark Ellison as editors.

1605 The following individuals were members of the committee while the specification 1.1 version was
1606 developed and approved by the technical committee: Guru Bhat, Jeff Bohren, Vaughn Bullard,
1607 Winston Bumpus, Fred Carter, Michael Clements, David Cox, Zulah Eckert, Mark Ellison, John
1608 Fuller, Tony Gullotta, Heather Kreger, Richard Landau, Frederico Maciel, Tom Maguire, David
1609 Melgar, Bryan Murray, Richard Nikula, Mark Peel, Mitsunori Satomi, Thomas Studwell, William
1610 Vambenepe, Kirk Wilson, 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	
standard	2005-03-09	Igor Sedukhin	
wd v 1.1	2005-10-27	Kirk Wilson	Addition of new capability: OperationMetric Creation of Appendix A
wd v 1.1 # 3	2005-11-03	Kirk Wilson	Addition of new Metric properties Version 1.1 namespace and file names added
wd v 1.1 #4	2005-11-04	Kirk Wilson	Addition of new capability: OperationOperationalStatus
wd v 1.1 #5	2005-11-16	Kirk Wilson	Restructure treatment of OperationMetrics and

Rev	Date	By Whom	
			OperationOperationalStatus Remove pseudo-UML diagrams Editorial corrections (result of F2F review)
wd v 1.1 #6	2005-11-28	Kirk Wilson	Additional editorial corrections Delete section 2.4 (UML "mind-map")
wd v 1.1 #7	2006-01-27	Kirk Wilson	Schema syntax check. Added acknowledgements
Final wd	2006-01-27	Kirk Wilson	Prepare final WD for TC vote
cd v 1.1 #1	2006-02-23	Kirk Wilson	Committee Draft candidate

1612

1613

Appendix C. Notices

1614 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
1615 that might be claimed to pertain to the implementation or use of the technology described in this
1616 document or the extent to which any license under such rights might or might not be available;
1617 neither does it represent that it has made any effort to identify any such rights. Information on
1618 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
1619 website. Copies of claims of rights made available for publication and any assurances of licenses
1620 to be made available, or the result of an attempt made to obtain a general license or permission
1621 for the use of such proprietary rights by implementors or users of this specification, can be
1622 obtained from the OASIS Executive Director.

1623 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
1624 applications, or other proprietary rights which may cover technology that may be required to
1625 implement this specification. Please address the information to the OASIS Executive Director.

1626 Copyright © OASIS Open 2003-2006. *All Rights Reserved.*

1627 This document and translations of it may be copied and furnished to others, and derivative works
1628 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
1629 published and distributed, in whole or in part, without restriction of any kind, provided that the
1630 above copyright notice and this paragraph are included on all such copies and derivative works.
1631 However, this document itself does not be modified in any way, such as by removing the
1632 copyright notice or references to OASIS, except as needed for the purpose of developing OASIS
1633 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
1634 Property Rights document must be followed, or as required to translate it into languages other
1635 than English.

1636 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
1637 successors or assigns.

1638 This document and the information contained herein is provided on an "AS IS" basis and OASIS
1639 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
1640 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
1641 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
1642 PARTICULAR PURPOSE.

1643

Appendix D. XML Schemas

```

1645 <?xml version="1.0" encoding="utf-8"?>
1646 <xs:schema targetNamespace="http://docs.oasis-open.org/wsdm/mows-2.xsd"
1647     xmlns:mows="http://docs.oasis-open.org/wsdm/mows-2.xsd"
1648     xmlns:muws2="http://docs.oasis-open.org/wsdm/muws2-2.xsd"
1649     xmlns:muws1="http://docs.oasis-open.org/wsdm/muws1-2.xsd"
1650     xmlns:wsa="http://www.w3.org/2005/08/addressing"
1651     xmlns:xs="http://www.w3.org/2001/XMLSchema"
1652     elementFormDefault="qualified" attributeFormDefault="unqualified">
1653
1654     <xs:import namespace="http://docs.oasis-open.org/wsdm/muws1-2.xsd"
1655 schemaLocation="http://docs.oasis-open.org/wsdm/muws1-2.xsd"/>
1656     <xs:import namespace="http://docs.oasis-open.org/wsdm/muws2-2.xsd"
1657 schemaLocation="http://docs.oasis-open.org/wsdm/muws2-2.xsd"/>
1658     <xs:import namespace="http://www.w3.org/2005/08/addressing"
1659 schemaLocation="http://www.w3.org/2005/08/addressing/ws-addr.xsd"/>
1660
1661     <!-- MOWS::ManageabilityReferences -->
1662     <xs:element name="GetManageabilityReferences"/>
1663     <xs:element name="GetManageabilityReferencesResponse">
1664         <xs:complexType>
1665             <xs:sequence>
1666                 <xs:element ref="muws1:ManageabilityEndpointReference"
1667                     maxOccurs="unbounded"/>
1668             </xs:sequence>
1669         </xs:complexType>
1670     </xs:element>
1671
1672     <!-- MOWS::Identification -->
1673     <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
1674     <xs:element name="EndpointDescriptions">
1675         <xs:complexType>
1676             <xs:sequence>
1677                 <xs:element name="description" type="xs:anyURI"
1678                     minOccurs="0" maxOccurs="unbounded"/>
1679             </xs:sequence>
1680             <xs:anyAttribute namespace="##other" processContents="lax"/>
1681         </xs:complexType>
1682     </xs:element>
1683
1684     <!-- Operation Properties Attribute Group -->
1685     <xs:attributeGroup name="OperationNameGroup">
1686         <xs:attribute name="operationName" type="xs:NCName" use="required"/>
1687         <xs:attribute name="portType" type="xs:QName" use="optional"/>
1688     </xs:attributeGroup>
1689
1690     <!-- MOWS Operation Operation Status Type -->
1691     <xs:complexType name="OperationOperationalStatusType">
1692         <xs:sequence>
1693             <xs:element ref="muws2:OperationalStatus" />
1694             <xs:any namespace="##any" processContents="lax"
1695                 minOccurs="0" maxOccurs="unbounded"/>

```

```

1696         </xs:sequence>
1697         <xs:attributeGroup ref="OperationNameGroup"/>
1698         <xs:anyAttribute namespace="##other" processContents="lax"/>
1699     </xs:complexType>
1700
1701     <xs:element name="OperationOperationalStatus"
1702               type="OperationOperationalStatusType" />
1703
1704     <!-- MOWS::Metrics -->
1705     <xs:complexType name="IntegerCounter">
1706         <xs:simpleContent>
1707             <xs:extension base="xs:nonNegativeInteger">
1708                 <xs:attributeGroup ref="muws2:MetricAttributes"/>
1709                 <xs:anyAttribute namespace="##other" processContents="lax"/>
1710             </xs:extension>
1711         </xs:simpleContent>
1712     </xs:complexType>
1713
1714     <xs:complexType name="DurationMetric">
1715         <xs:simpleContent>
1716             <xs:extension base="xs:duration">
1717                 <xs:attributeGroup ref="muws2:MetricAttributes"/>
1718                 <xs:anyAttribute namespace="##other" processContents="lax"/>
1719             </xs:extension>
1720         </xs:simpleContent>
1721     </xs:complexType>
1722
1723     <xs:complexType name="OperationIntegerCounter">
1724         <xs:simpleContent>
1725             <xs:extension base="mows:IntegerCounter">
1726                 <xs:attributeGroup ref="OperationNameGroup"/>
1727             </xs:extension>
1728         </xs:simpleContent>
1729     </xs:complexType>
1730
1731     <xs:complexType name="OperationDurationMetric">
1732         <xs:simpleContent>
1733             <xs:extension base="mows:DurationMetric">
1734                 <xs:attributeGroup ref="OperationNameGroup"/>
1735             </xs:extension>
1736         </xs:simpleContent>
1737     </xs:complexType>
1738
1739     <xs:element name="NumberOfRequests" type="mows:IntegerCounter"/>
1740     <xs:element name="NumberOfSuccessfulRequests" type="mows:IntegerCounter"/>
1741     <xs:element name="NumberOfFailedRequests" type="mows:IntegerCounter"/>
1742     <xs:element name="ServiceTime" type="mows:DurationMetric"/>
1743     <xs:element name="MaxResponseTime" type="mows:DurationMetric"/>
1744     <xs:element name="LastResponseTime" type="mows:DurationMetric"/>
1745     <xs:element name="MaxRequestSize" type="mows:IntegerCounter"/>
1746     <xs:element name="LastRequestSize" type="mows:IntegerCounter"/>
1747     <xs:element name="MaxReponseSize" type="mows:IntegerCounter"/>
1748     <xs:element name="LastReponseSize" type="mows:IntegerCounter"/>
1749
1750     <!-- MOWS::Operation Metrics -->

```

```

1751 <xs:complexType name="OperationMetricType">
1752   <xs:sequence>
1753     <xs:element ref="NumberOfRequests" minOccurs="0"/>
1754     <xs:element ref="NumberOfSuccessfulRequests"
1755       minOccurs="0"/>
1756     <xs:element ref="NumberOfFailedRequests" minOccurs="0"/>
1757     <xs:element ref="ServiceTime" minOccurs="0"/>
1758     <xs:element ref="MaxResponseTime" minOccurs="0"/>
1759     <xs:element ref="LastResponseTime" minOccurs="0"/>
1760     <xs:element ref="MaxRequestSize" minOccurs="0"/>
1761     <xs:element ref="LastRequestSize" minOccurs="0"/>
1762     <xs:element ref="MaxResponseSize" minOccurs="0"/>
1763     <xs:element ref="LastResponseSize" minOccurs="0"/>
1764     <xs:any namespace="##any" processContents="lax"
1765       minOccurs="0" maxOccurs="unbounded"/>
1766   </xs:sequence>
1767   <xs:attributeGroup ref="OperationNameGroup" />
1768   <xs:anyAttribute namespace="##other" processContents="lax"/>
1769 </xs:complexType>
1770
1771 <xs:element name="OperationMetrics" type="OperationMetricType"/>
1772
1773 <!-- MOWS::OperationalState -->
1774 <xs:complexType name="OperationalStateType">
1775   <xs:complexContent>
1776     <xs:extension base="muws2:StateType"/>
1777   </xs:complexContent>
1778 </xs:complexType>
1779 <xs:element name="UpState">
1780   <xs:complexType>
1781     <xs:complexContent>
1782       <xs:restriction base="mows:OperationalStateType"/>
1783     </xs:complexContent>
1784   </xs:complexType>
1785 </xs:element>
1786 <xs:element name="IdleState">
1787   <xs:complexType>
1788     <xs:complexContent>
1789       <xs:restriction base="mows:OperationalStateType">
1790         <xs:sequence>
1791           <xs:element ref="mows:UpState"/>
1792         </xs:sequence>
1793       </xs:restriction>
1794     </xs:complexContent>
1795   </xs:complexType>
1796 </xs:element>
1797 <xs:element name="BusyState">
1798   <xs:complexType>
1799     <xs:complexContent>
1800       <xs:restriction base="mows:OperationalStateType">
1801         <xs:sequence>
1802           <xs:element ref="mows:UpState"/>
1803         </xs:sequence>
1804       </xs:restriction>
1805     </xs:complexContent>

```

```

1806         </xs:complexType>
1807     </xs:element>
1808     <xs:element name="DownState">
1809         <xs:complexType>
1810             <xs:complexContent>
1811                 <xs:restriction base="mows:OperationalStateType"/>
1812             </xs:complexContent>
1813         </xs:complexType>
1814     </xs:element>
1815     <xs:element name="StoppedState">
1816         <xs:complexType>
1817             <xs:complexContent>
1818                 <xs:restriction base="mows:OperationalStateType">
1819                     <xs:sequence>
1820                         <xs:element ref="mows:DownState"/>
1821                     </xs:sequence>
1822                 </xs:restriction>
1823             </xs:complexContent>
1824         </xs:complexType>
1825     </xs:element>
1826     <xs:element name="CrashedState">
1827         <xs:complexType>
1828             <xs:complexContent>
1829                 <xs:restriction base="mows:OperationalStateType">
1830                     <xs:sequence>
1831                         <xs:element ref="mows:DownState"/>
1832                     </xs:sequence>
1833                 </xs:restriction>
1834             </xs:complexContent>
1835         </xs:complexType>
1836     </xs:element>
1837     <xs:element name="SaturatedState">
1838         <xs:complexType>
1839             <xs:complexContent>
1840                 <xs:restriction base="mows:OperationalStateType">
1841                     <xs:sequence>
1842                         <xs:element ref="mows:DownState"/>
1843                     </xs:sequence>
1844                 </xs:restriction>
1845             </xs:complexContent>
1846         </xs:complexType>
1847     </xs:element>
1848
1849     <xs:element name="CurrentOperationalState" type="mows:OperationalStateType"/>
1850     <xs:element name="LastOperationalStateTransition"
1851         type="muws2:StateTransitionType"/>
1852
1853     <!-- MOWS::RequestProcessingState -->
1854     <xs:complexType name="RequestProcessingStateType">
1855         <xs:complexContent>
1856             <xs:extension base="muws2:StateType"/>
1857         </xs:complexContent>
1858     </xs:complexType>
1859     <xs:element name="RequestReceivedState">
1860         <xs:complexType>

```

```

1861         <xs:complexContent>
1862             <xs:restriction base="mows:RequestProcessingStateType"/>
1863         </xs:complexContent>
1864     </xs:complexType>
1865 </xs:element>
1866 <xs:element name="RequestProcessingState">
1867     <xs:complexType>
1868         <xs:complexContent>
1869             <xs:restriction base="mows:RequestProcessingStateType"/>
1870         </xs:complexContent>
1871     </xs:complexType>
1872 </xs:element>
1873 <xs:element name="RequestCompletedState">
1874     <xs:complexType>
1875         <xs:complexContent>
1876             <xs:restriction base="mows:RequestProcessingStateType"/>
1877         </xs:complexContent>
1878     </xs:complexType>
1879 </xs:element>
1880 <xs:element name="RequestFailedState">
1881     <xs:complexType>
1882         <xs:complexContent>
1883             <xs:restriction base="mows:RequestProcessingStateType"/>
1884         </xs:complexContent>
1885     </xs:complexType>
1886 </xs:element>
1887 <xs:complexType name="MessageContentNotIncludedFlag"/>
1888 <xs:simpleType name="MessageSizeUnitType">
1889     <xs:restriction base="xs:string">
1890         <xs:enumeration value="bit"/>
1891         <xs:enumeration value="byte"/>
1892         <xs:enumeration value="word"/>
1893         <xs:enumeration value="dword"/>
1894         <xs:enumeration value="qword"/>
1895     </xs:restriction>
1896 </xs:simpleType>
1897 <xs:complexType name="MessageContentSizeType">
1898     <xs:simpleContent>
1899         <xs:extension base="xs:positiveInteger">
1900             <xs:attribute name="Unit"
1901                 type="mows:MessageSizeUnitType" use="required"/>
1902             <xs:anyAttribute namespace="##other" processContents="lax"/>
1903         </xs:extension>
1904     </xs:simpleContent>
1905 </xs:complexType>
1906 <xs:complexType name="MessageContentType">
1907     <xs:sequence>
1908         <xs:element name="Size"
1909             type="mows:MessageContentSizeType" minOccurs="0"/>
1910         <xs:choice>
1911             <xs:element name="NotIncluded"
1912                 type="mows:MessageContentNotIncludedFlag"/>
1913             <xs:element name="Text" type="xs:string"/>
1914             <xs:element name="Binary" type="xs:base64Binary"/>
1915             <xs:element name="Xml"

```

```

1916         type="mows:AnyXmlContentsType"/>
1917     </xs:choice>
1918     <xs:any namespace="##other" processContents="lax"
1919         minOccurs="0" maxOccurs="unbounded"/>
1920 </xs:sequence>
1921 <xs:anyAttribute namespace="##other" processContents="lax"/>
1922 </xs:complexType>
1923 <xs:complexType name="AnyXmlContentsType">
1924     <xs:sequence>
1925         <xs:any namespace="##any" processContents="lax"
1926             minOccurs="0" maxOccurs="unbounded"/>
1927     </xs:sequence>
1928     <xs:anyAttribute namespace="##any" processContents="lax"/>
1929 </xs:complexType>
1930 <xs:complexType name="MessageInformationType">
1931     <xs:sequence>
1932         <xs:element name="TransportInformation"
1933             type="mows:AnyXmlContentsType" minOccurs="0"/>
1934         <xs:element name="Message" type="mows:MessageContentType"/>
1935         <xs:any namespace="##any" processContents="lax"
1936             minOccurs="0" maxOccurs="unbounded"/>
1937     </xs:sequence>
1938     <xs:anyAttribute namespace="##any" processContents="lax"/>
1939 </xs:complexType>
1940 <xs:complexType name="RequestProcessingStateInformationType">
1941     <xs:sequence>
1942         <xs:element ref="muws2:StateTransition" maxOccurs="unbounded"/>
1943     </xs:sequence>
1944 </xs:complexType>
1945 <xs:element name="RequestProcessingNotification">
1946     <xs:complexType>
1947         <xs:sequence>
1948             <xs:element name="Request"
1949                 type="mows:MessageInformationType"
1950                 minOccurs="0"/>
1951             <xs:element name="Reply"
1952                 type="mows:MessageInformationType"
1953                 minOccurs="0"/>
1954             <xs:element name="StateInformation"
1955                 type="mows:RequestProcessingStateInformationType"/>
1956             <xs:any namespace="##any" processContents="lax"
1957                 minOccurs="0" maxOccurs="unbounded"/>
1958         </xs:sequence>
1959         <xs:attribute name="CurrentTime" type="xs:dateTime" use="required"/>
1960         <xs:anyAttribute namespace="##any" processContents="lax"/>
1961     </xs:complexType>
1962 </xs:element>
1963 <xs:simpleType name="IPv4AddressType">
1964     <xs:restriction base="xs:hexBinary">
1965         <xs:length value="8" fixed="true"/>
1966     </xs:restriction>
1967 </xs:simpleType>
1968 <xs:element name="IPv4Address" type="mows:IPv4AddressType"/>
1969 <xs:simpleType name="IPv6AddressType">
1970     <xs:restriction base="xs:hexBinary">

```

```

1971         <xs:length value="32" fixed="true"/>
1972     </xs:restriction>
1973 </xs:simpleType>
1974 <xs:element name="IPv6Address" type="mows:IPv6AddressType"/>
1975 <xs:simpleType name="TcplpDirectionType">
1976     <xs:restriction base="xs:string">
1977         <xs:enumeration value="to"/>
1978         <xs:enumeration value="from"/>
1979     </xs:restriction>
1980 </xs:simpleType>
1981 <xs:simpleType name="TcplpProtocolType">
1982     <xs:restriction base="xs:string">
1983         <xs:enumeration value="TCP"/>
1984         <xs:enumeration value="UDP"/>
1985     </xs:restriction>
1986 </xs:simpleType>
1987 <xs:element name="TcplpInfo">
1988     <xs:complexType>
1989         <xs:sequence>
1990             <xs:choice>
1991                 <xs:element ref="mows:IPv4Address"/>
1992                 <xs:element ref="mows:IPv6Address"/>
1993             </xs:choice>
1994             <xs:any namespace="##any" processContents="lax"
1995                 minOccurs="0" maxOccurs="unbounded"/>
1996         </xs:sequence>
1997         <xs:attribute name="Direction" type="mows:TcplpDirectionType"
1998             use="required"/>
1999         <xs:attribute name="Port" type="xs:positiveInteger" use="required"/>
2000         <xs:attribute name="Protocol" type="mows:TcplpProtocolType"
2001             use="required"/>
2002         <xs:anyAttribute namespace="##any" processContents="lax"/>
2003     </xs:complexType>
2004 </xs:element>
2005 </xs:schema>
2006
2007 <!--
2008             SCHEMA COPY Material
2009 Copy and paste element references below into the schema of a resource properties document.
2010 These references are provide to insure that the correct minOccurs/maxOccurs attributes are
2011 specified in a resource property document schema.
2012
2013 NOTE: You must import the MOWS schema namespace (mows).
2014
2015     ** Endpoint Identification Properties **
2016     <xs:element ref="mows:EndpointReference"/>
2017     <xs:element ref="mows:EndpointDescriptions" minOccurs="0"/>
2018
2019     ** MOWS Metric Properties **
2020     <xs:element ref="mows:NumberOfRequests" minOccurs="0"/>
2021     <xs:element ref="mows:NumberOfFailedRequests" minOccurs="0"/>
2022     <xs:element ref="mows:NumberOfSuccessfulRequests" minOccurs="0"/>
2023     <xs:element ref="mows:ServiceTime" minOccurs="0"/>
2024     <xs:element ref="mows:MaxResponseTime" minOccurs="0"/>
2025     <xs:element ref="mows>LastResponseTime" minOccurs="0"/>

```

```
2026 <xs:element ref="mows:MaxRequestSize" minOccurs="0"/>
2027 <xs:element ref="mows:LastRequestSize" minOccurs="0"/>
2028 <xs:element ref="mows:MaxResponseSize" minOccurs="0"/>
2029 <xs:element ref="mows:LastResponseSize" minOccurs="0"/>
2030
2031 **- MOWS Operation Metric Property **
2032 <xs:element ref="mows:OperationMetrics" minOccurs="0" maxOccurs="unbounded"/>
2033
2034 ** MOWS Operation Operational Status Property **
2035 <xs:element ref="mows:OperationOperationalStatus"
2036         minOccurs="0" maxOccurs="unbounded"/>
2037
2038 ** Operational State Properties **
2039 <xs:element ref="mows:CurrentOperationalState"/>
2040 <xs:element ref="mows:LastOperationalStateTransition" minOccurs="0"/>
2041
2042 -->
2043
```

2044

Appendix E. WSDL elements

```
2045 <?xml version="1.0" encoding="utf-8"?>
2046 <w:definitions xmlns:w="http://schemas.xmlsoap.org/wsdl/"
2047     xmlns:xs="http://www.w3.org/2001/XMLSchema"
2048     xmlns:wsrp="http://docs.oasis-open.org/wsrp/rp-2"
2049     xmlns:mows="http://docs.oasis-open.org/wsdm/mows-2.xsd"
2050     xmlns:mowsw="http://docs.oasis-open.org/wsdm/mows-2.wsdl"
2051     targetNamespace="http://docs.oasis-open.org/wsdm/mows-2.wsdl">
2052
2053     <w:types>
2054         <xs:import namespace="http://docs.oasis-open.org/wsdm/mows-2.xsd"
2055 schemaLocation="http://docs.oasis-open.org/wsdm/mows-2.xsd"/>
2056     </w:types>
2057
2058     <w:message name="GetManageabilityReferencesRequest">
2059         <w:part name="body" element="mows:GetManageabilityReferences"/>
2060     </w:message>
2061     <w:message name="GetManageabilityReferencesResponse">
2062         <w:part name="body"
2063             element="mows:GetManageabilityReferencesResponse"/>
2064     </w:message>
2065
2066 </w:definitions>
2067
2068 <!--
2069             WSDL COPY Material
2070 Cut and Paste the operation specification below into a portType definition of the WSDL
2071 documents of a web service.
2072
2073 NOTE: You must import the MOWS WSDL (mowsw).
2074
2075     <operation name="GetManageabilityReferences">
2076         <input name="GetManageabilityReferencesRequest"
2077             message="mowsw:GetManageabilityReferencesRequest"/>
2078         <output name="GetManageabilityReferencesResponse"
2079             message="mowsw:GetManageabilityReferencesResponse"/>
2080     </operation>
2081
2082 -->
2083
```

Appendix F. Notification topic spaces

```

2085 <?xml version="1.0" encoding="utf-8"?>
2086 <wstop:TopicSpace name="MOWS"
2087     targetNamespace="http://docs.oasis-open.org/wsdm/mowse-2.xml"
2088     xmlns:muws2="http://docs.oasis-open.org/wsdm/muws2-2.xsd"
2089     xmlns:muws1="http://docs.oasis-open.org/wsdm/muws1-2.xsd"
2090     xmlns:mows="http://docs.oasis-open.org/wsdm/mows-2.xsd"
2091     xmlns:wstop="http://docs.oasis-open.org/wsn/t-1">
2092
2093 <wstop:Topic name="IdentificationCapability" messageTypes="muws1:ManagementEvent"/>
2094 <wstop:Topic name="MetricsCapability" messageTypes="muws1:ManagementEvent"/>
2095 <wstop:Topic name="OperationMetricsCapability" messageTypes="muws1:ManagementEvent"/>
2096 <wstop:Topic name="OperationalStateCapability" messageTypes="muws1:ManagementEvent"/>
2097 <wstop:Topic name="OperationalStatusCapability"
2098     messageTypes="muws1:ManagementEvent"/>
2099 <wstop:Topic name="OperationOperationalStatusCapability"
2100     messageTypes="muws1:ManagementEvent"/>
2101 <wstop:Topic name="RequestProcessingStateCapability"
2102     messageTypes="muws1:ManagementEvent"/>
2103
2104
2105 <wstop:Topic name="RequestProcessingObservations"
2106     messageTypes="muws1:ManagementEvent">
2107     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
2108 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
2109 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
2110     </wstop:MessagePattern>
2111     <wstop:Topic name="RequestReceived"
2112         messageTypes="muws1:ManagementEvent">
2113         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
2114 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
2115 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
2116         </wstop:MessagePattern>
2117         </wstop:Topic>
2118     <wstop:Topic name="RequestProcessing"
2119         messageTypes="muws1:ManagementEvent">
2120         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
2121 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
2122 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
2123         </wstop:MessagePattern>
2124         </wstop:Topic>
2125     <wstop:Topic name="RequestCompleted"
2126         messageTypes="muws1:ManagementEvent">
2127         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
2128 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
2129 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
2130         </wstop:MessagePattern>
2131         </wstop:Topic>
2132     <wstop:Topic name="RequestFailed"
2133         messageTypes="muws1:ManagementEvent">
2134     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">

```

```

2135 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
2136 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
2137     </wstop:MessagePattern>
2138     </wstop:Topic>
2139     <wstop:Topic name="Digest"
2140         messageTypes="muws1:ManagementEvent">
2141         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
2142 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
2143 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
2144         </wstop:MessagePattern>
2145         </wstop:Topic>
2146 </wstop:Topic>
2147
2148 <wstop:Topic name="RequestProcessingObservationsWithAttachments"
2149     messageTypes="muws1:ManagementEvent">
2150     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
2151 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
2152 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
2153     </wstop:MessagePattern>
2154     <wstop:Topic name="RequestReceived"
2155         messageTypes="muws1:ManagementEvent">
2156         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
2157 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
2158 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
2159         </wstop:MessagePattern>
2160         </wstop:Topic>
2161         <wstop:Topic name="RequestProcessing"
2162             messageTypes="muws1:ManagementEvent">
2163             <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
2164 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
2165 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
2166             </wstop:MessagePattern>
2167             </wstop:Topic>
2168             <wstop:Topic name="RequestCompleted"
2169                 messageTypes="muws1:ManagementEvent">
2170                 <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
2171 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
2172 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
2173                 </wstop:MessagePattern>
2174                 </wstop:Topic>
2175                 <wstop:Topic name="RequestFailed"
2176                     messageTypes="muws1:ManagementEvent">
2177                     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
2178 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
2179 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
2180                     </wstop:MessagePattern>
2181                     </wstop:Topic>
2182                     <wstop:Topic name="Digest"
2183                         messageTypes="muws1:ManagementEvent">
2184                         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
2185 //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
2186 and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
2187                         </wstop:MessagePattern>
2188                         </wstop:Topic>
2189 </wstop:Topic>

```

