



1

2

3

4

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

5

OASIS-Standard, 9 March 2005

6

**Document identifier:**

7

wsdm-mows-1.0

8

**Location:**

9

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

10

**Editors:**

11

*Igor Sedukhin, Computer Associates* <[igor.sedukhin@ca.com](mailto:igor.sedukhin@ca.com)>

12

**Abstract:**

13

14

15

16

17

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.

18

**Status:**

19

This document is an OASIS standard.

20

21

22

23

24

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

25

26

27

28

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/>).

29

30

The errata document for this specification is maintained at: <http://docs.oasis-open.org/wsdm/2004/12/wsdm-mows-1.0-errata.pdf>

---

31 **Table of Contents**

32 1 Introduction ..... 3  
33 1.1 Terminology..... 3  
34 1.2 Notational conventions ..... 4  
35 2 Architecture ..... 5  
36 2.1 In-band and Out-of-band Manageability ..... 6  
37 2.2 Application to Resources Exposed as Web Services ..... 6  
38 2.3 Self-Management ..... 7  
39 2.4 Formal Representation of the Architecture ..... 8  
40 3 Managing Web Services ..... 9  
41 3.1 Responsibilities of the Implementations of the Manageability Endpoints ..... 9  
42 3.2 Manageability at the Web service level..... 10  
43 3.3 Using manageability of Web services endpoints ..... 10  
44 4 Security Considerations ..... 12  
45 4.1 Additional security considerations when managing Web services ..... 12  
46 5 Web service manageability capabilities ..... 14  
47 5.1 Common manageability capabilities..... 15  
48 5.1.1 Manageability References ..... 15  
49 5.2 Web service endpoint manageability capabilities ..... 16  
50 5.2.1 Identity ..... 16  
51 5.2.2 Identification..... 17  
52 5.2.3 Metrics ..... 18  
53 5.2.4 Operational State ..... 21  
54 5.2.5 Operational Status ..... 24  
55 5.2.6 Request Processing State ..... 24  
56 6 References..... 35  
57 6.1 Normative ..... 35  
58 6.2 Non-normative..... 35  
59 Appendix A. Acknowledgments ..... 37  
60 Appendix B. Revision History ..... 38  
61 Appendix C. Notices ..... 39  
62 Appendix D. XML Schemas..... 40  
63 Appendix E. WSDL elements ..... 47  
64 Appendix F. Notification topic spaces ..... 48

65

---

## 66 1 Introduction

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

75

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

79

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

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

93

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

96

### 97 1.1 Terminology

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

101

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

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

105

## 106 1.2 Notational conventions

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

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

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

126

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

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

140

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

147

148

## 2 Architecture

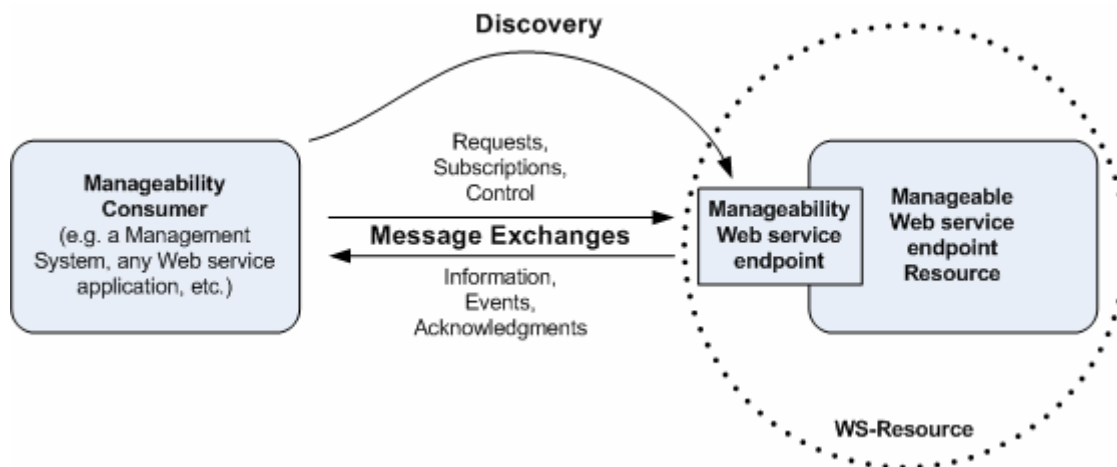
149

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

156

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

166



167

168

**Figure 1.** Management of Web services concepts

169

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

173

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

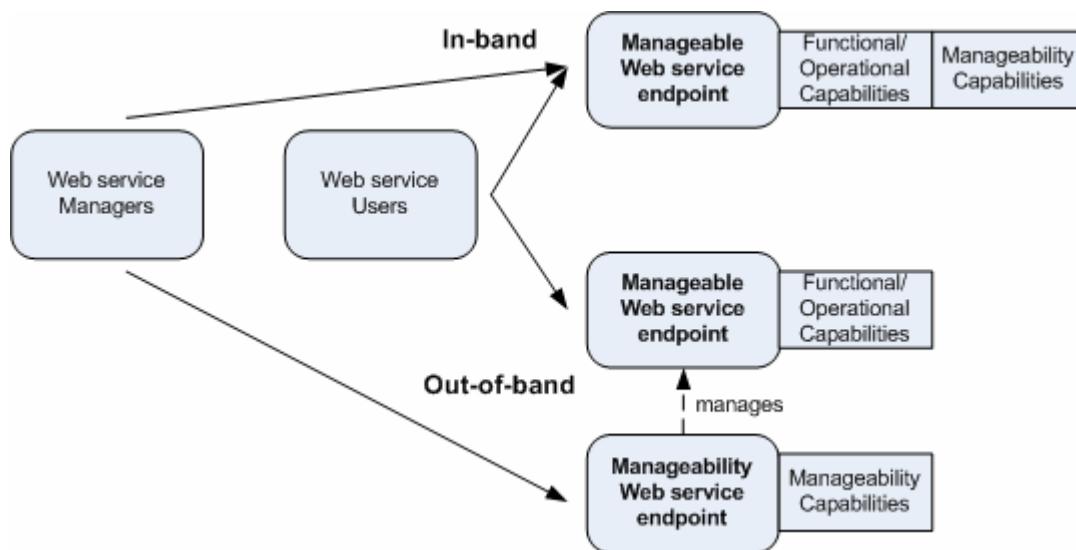
176

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

181 **2.1 In-band and Out-of-band Manageability**

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

189



190

191

192

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

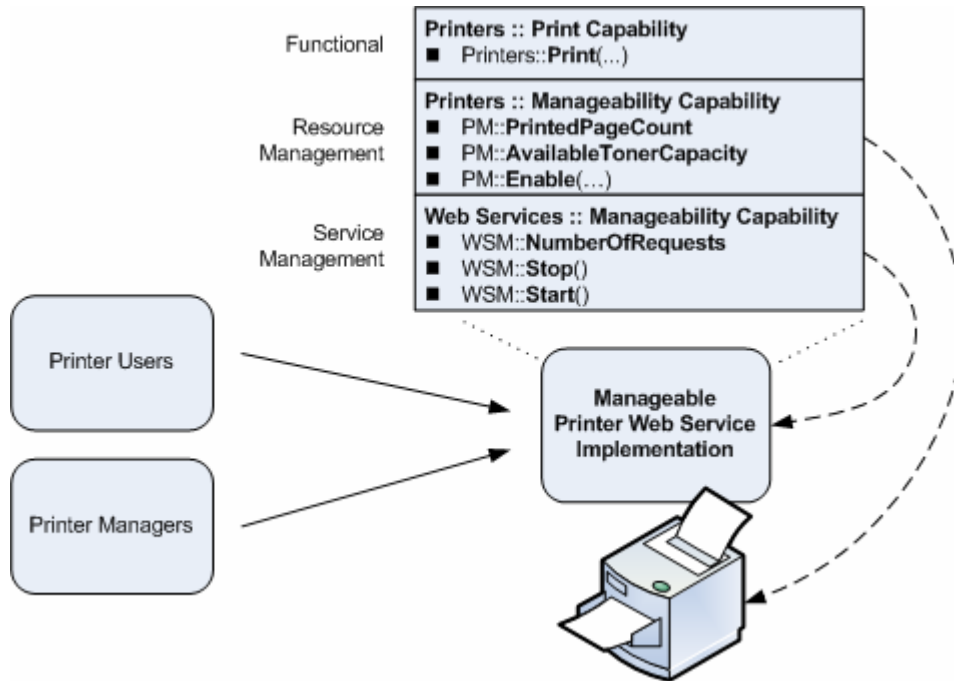
193 **2.2 Application to Resources Exposed as Web Services**

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

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

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

208

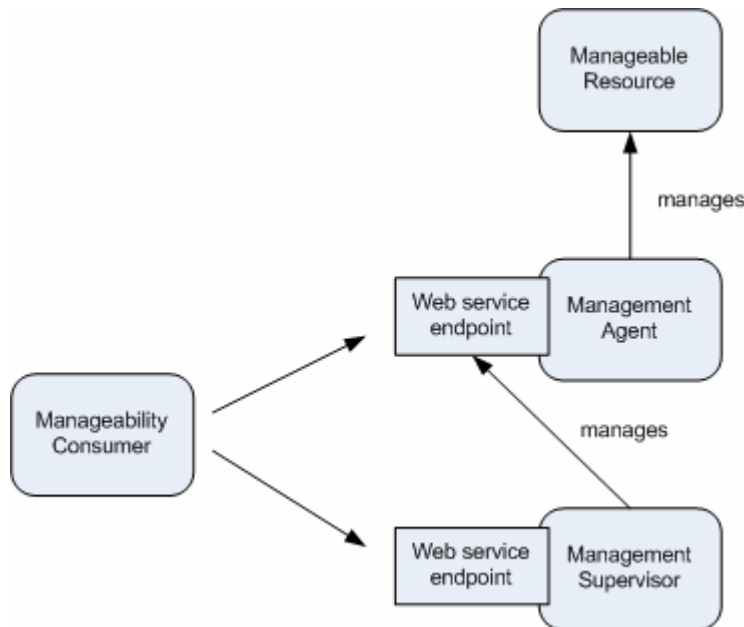


209  
210

**Figure 3.**Application to resources exposed as Web services

211 **2.3 Self-Management**

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



217  
218  
219

**Figure 4.**Applying MOWS to MUWS





---

## 227 **3 Managing Web Services**

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

235

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

246

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

### 249 **3.1 Responsibilities of the Implementations of the Manageability** 250 **Endpoints**

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

256

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

262

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

267

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

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

277

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

## 281 **3.2 Manageability at the Web service level**

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

289

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

## 295 **3.3 Using manageability of Web services endpoints**

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

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

321  
322  
323  
324

- 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.

---

## 325 4 Security Considerations

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

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

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

350

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

356

### 357 4.1 Additional security considerations when managing Web 358 services

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

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

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

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

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

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

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

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

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

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

392

---

## 5 Web service manageability capabilities

393

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

396

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

399

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

404

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

407

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

408

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

411

## 412 5.1 Common manageability capabilities

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

### 415 5.1.1 Manageability References

416 This capability is identified by the following URI:

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

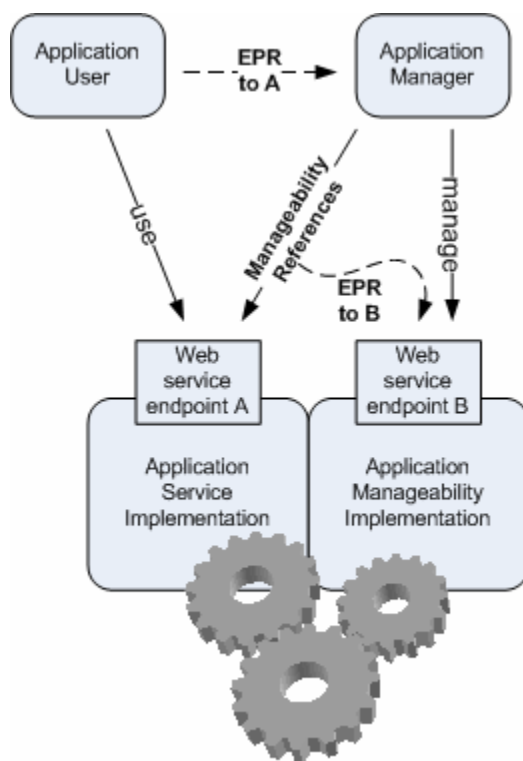
418

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

425

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

431



432

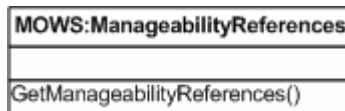
433

**Figure 6.** Use of Manageability References capability

434

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

437



438

439

Figure 7. Manageability References capability model

440

### 441 5.1.1.1 Operations

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

443

#### 444 5.1.1.1.1 GetManageabilityReferences

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

447

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

449

```
450 <GetManageabilityReferences/>
```

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

453

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

456

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

462

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

465

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

470

## 471 5.2 Web service endpoint manageability capabilities

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

### 473 5.2.1 Identity

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



## 476 5.2.2 Identification

477 This capability is identified by the following URI:

478 **http://docs.oasis-open.org/wsdm/2004/12/mows/capabilities/Identification**

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

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

482

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

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

489

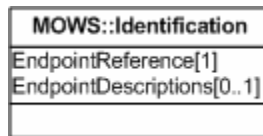
490 Note that, in contrast, the MUWS **Identity** capability and semantics are consistent with the  
491 following definition (from the Webster dictionary).

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

494

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

498



499

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

501

### 502 5.2.2.1 Properties

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

505

```
506 <EndpointReference>wsa:EndpointReferenceType</EndpointReference>  
507 <EndpointDescriptions><description>xs:anyURI</description>*</EndpointDescriptions>?
```

508

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

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

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

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

### 520    **5.2.2.2 Events**

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

```
523
524    <wstop:Topic name="IdentificationCapability" messageTypes="muws-xs1:ManagementEvent"/>
```

525
 526    **mows-events:IdentificationCapability** is a topic on which management events related to this  
 527    manageability capability SHOULD be emitted.

### 529    **5.2.3 Metrics**

530    This capability is identified by the following URI:

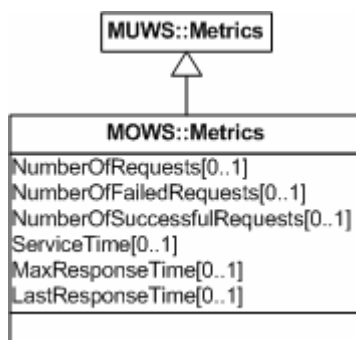
531    <http://docs.oasis-open.org/wsdm/2004/12/mows/capabilities/Metrics>

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

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

535
 536    The Web service endpoint's manageable metrics capability is represented by the **Metrics** UML  
 537    model class. The name of the class identifies the semantics of this capability.

538



539

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

541

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

545

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

549

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

### 553 **5.2.3.1 Information markup declarations**

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

556

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

565

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

567

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

576

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

### 578 **5.2.3.2 Properties**

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

581

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

588

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

592

- Is *Mutable*

593

- Is not *Modifiable*

594

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

595

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

597

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

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

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

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

615

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

618

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

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

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

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

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

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

639

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

646

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

### 651 5.2.3.3 Events

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

654

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

656

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

659

### 660 5.2.4 Operational State

661 This capability is identified by the following URI:

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

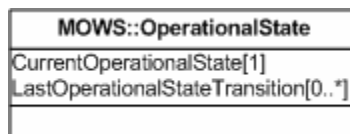
663 All properties, operations and events defined for this capability have the following metadata:

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

666

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

670



671

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

673

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

678

#### 679 5.2.4.1 Information markup declarations

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

- 683 ▪ **UpState**

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

- 687 ▪ **DownState**

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

691     ▪ **BusyState**

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

695     ▪ **IdleState**

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

699     ▪ **StoppedState**

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

703     ▪ **CrashedState**

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

707     ▪ **SaturatedState**

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

711

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

716

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

718

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

724

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

727

728     ▪ A substate of the operational state MUST be declared according to the following rules.

729         ○ An XML element is declared with a QName which identifies the desired substate  
730 semantics, for example my-app:DatabaseCleanupState

731         ○ The contents of the XML element MUST be the only element which corresponds  
732 to the generalized state, for example mows-xs:StoppedState

733

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

736

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

744

#### 745 5.2.4.2 Properties

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

748

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

753

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

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

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

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

763

#### 764 5.2.4.3 Events

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

767

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

770

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

773

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

778

## 779 5.2.5 Operational Status

780 This capability is identified by the following URI:

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

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

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

785

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

789

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

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

801

### 802 5.2.5.1 Events

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

805

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

808

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

811

## 812 5.2.6 Request Processing State

813 This capability is identified by the following URI:

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

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

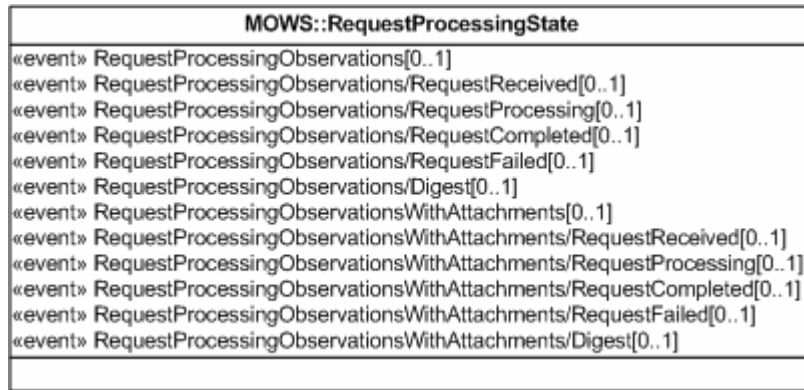
- 816     ▪ `<muws-xs2:Capability>http://docs.oasis-`
- 817         `open.org/wsdm/2004/12/mows/capabilities/RequestProcessingState</muws-`
- 818         `xs2:Capability>`

819

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



823



824

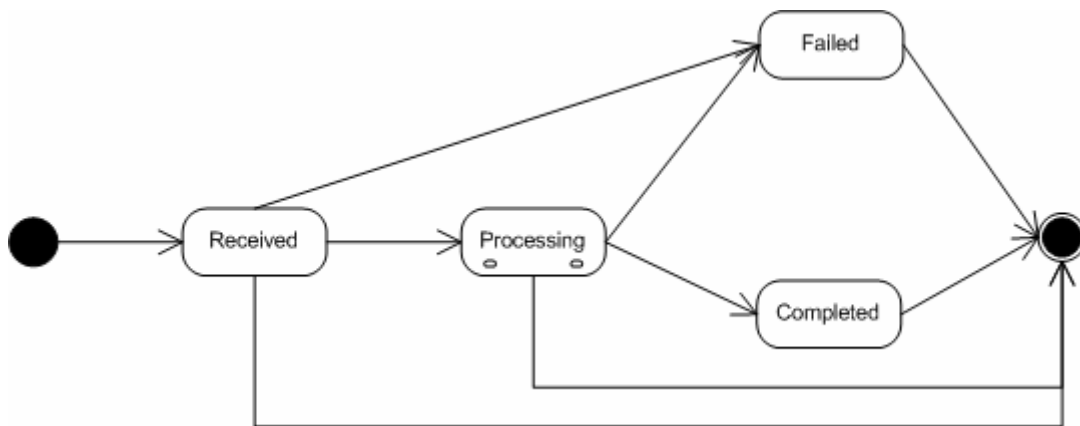
825

Figure 11. Endpoint request processing state manageability capability model

826

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

830



831

832

Figure 12. Request processing states

833

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

837

### 838 5.2.6.1 Information markup declarations

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

842

- **RequestReceivedState**

843

844

845

846

847

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

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

864

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

869

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

871

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

877

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

880

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

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

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

886

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

889

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

896

## 897 5.2.6.2 Events

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

899

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

903

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

```

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

```

1005 **mows-events:RequestProcessingStateCapability** is a topic on which management events  
1006 related to this manageability capability SHOULD be emitted.

1007 **mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations**  
1008 indicates availability of any information about the processing of any request by the Web service  
1009 endpoint (Figure 12) as observed by the implementation of a manageable Web service.

1010 The notification message for this topic MUST contain at most one  
1011 **RequestProcessingNotification** element (defined in §5.2.6.2.1). The MUWS  
1012 management event MUST also declare the event situation category with the muws-  
1013 xs2:ReportSituation element and the severity value “1” (Informational). It is recommended  
1014 to subscribe to these topics with proper preconditions and selectors using expressions  
1015 against the contents of the RequestProcessingNotification element.

1016 **mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations/mows-**  
1017 **events:RequestReceived** indicates that a request was received by the Web service endpoint  
1018 being managed (Received state on the Figure 12). The notification message format for this topic  
1019 is the same as the notification message format for the mows-events:ManageableEndpoint/mows-  
1020 events:RequestProcessingObservations topic. This is a state change event and therefore  
1021 notification messages MUST contain exactly one muws-xs2:StateTransition element inside of the  
1022 RequestProcessingNotification/StateInformation element.

1023 **mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations/mows-**  
1024 **events:RequestProcessing** indicates that a request is being processed by the Web service  
1025 endpoint being managed (Processing state on the Figure 12). The notification message format for  
1026 this topic is the same as the notification message format for the mows-  
1027 events:ManageableEndpoint/mows-events:RequestProcessingObservations topic. This is a state  
1028 change event and therefore notification messages MUST contain exactly one muws-  
1029 xs2:StateTransition element inside of the RequestProcessingNotification/StateInformation  
1030 element.

1031 **mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations/mows-**  
1032 **events:RequestCompleted** indicates that a request was successfully completed by the Web  
1033 service endpoint being managed (Completed state on the Figure 12). The notification message  
1034 format for this topic is the same as the notification message format for the mows-  
1035 events:ManageableEndpoint/mows-events:RequestProcessingObservations topic. This is a state  
1036 change event and therefore notification messages MUST contain exactly one muws-  
1037 xs2:StateTransition element inside of the RequestProcessingNotification/StateInformation  
1038 element.

1039 **mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations/mows-**  
1040 **events:RequestFailed** indicates that a request was failed (not successfully completed) by the  
1041 Web service endpoint being managed (Failed state on the Figure 12). The notification message  
1042 format for this topic is the same as the notification message format for the mows-  
1043 events:ManageableEndpoint/mows-events:RequestProcessingObservations topic. This is a state  
1044 change event and therefore notification messages MUST contain exactly one muws-  
1045 xs2:StateTransition element inside of the RequestProcessingNotification/StateInformation  
1046 element.

1047 **mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations/mows-**  
1048 **events:Digest** indicates availability of summary information about a request processed by the  
1049 Web service endpoint being managed. The notification message format for this topic is the same  
1050 as the notification message format for the mows-events:ManageableEndpoint/mows-  
1051 events:RequestProcessingObservations topic. This is a digest event and therefore notification  
1052 messages MUST contain one or more muws-xs2:StateTransition elements inside of the  
1053 RequestProcessingNotification/StateInformation element. Each muws-xs2:StateTransition  
1054 element describes a state transition which occurred with that one request which this summary  
1055 notification is informing about. Each state transition information element carries an attribute  
1056 indicating the time when that particular transition occurred. Using this information the  
1057 manageability consumer can reconstruct the sequence of events with regards to the request.

1058 **mows-events:ManageableEndpoint/mows-**  
1059 **events:RequestProcessingObservationsWithAttachments** topic and all of its subtopics are  
1060 defined exactly as the mows-events:ManageableEndpoint/mows-  
1061 events:RequestProcessingObservations topic and its respective subtopics, except that the  
1062 notification messages **MUST** include attachments (if any) of the request and reply messages sent  
1063 to/from the Web service endpoint being managed.

1064 The notification message format for this topic and all of its subtopics is the same as the  
1065 notification message format for the mows-events:ManageableEndpoint/mows-  
1066 events:RequestProcessingObservations topic, except that attachments may be sent  
1067 along with the message. The precise mechanism of sending the attachment is dependent  
1068 on 1) the binding of the notification consumer Web service endpoint **[WS-N]** and 2) the  
1069 binding of the Web service endpoint being managed.

1070

1071 The mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations/mows-  
1072 events:RequestProcessing topic **MAY** be extended with custom subtopics in order to represent  
1073 custom request processing substates of the Processing compound state (Figure 12).

1074

1075 Note that the result of the message pattern XPath expressions in the topic declarations above is  
1076 the XML nodeset **[XPath]** of the notification messages that are sent inside of the S:Body element  
1077 or the wsnt:Notify element **[WS-N]**.

1078

1079 Note that for the XPath expressions defined here the prefix-to-namespace mapping context  
1080 **MUST** include all prefixes which appear in the XPath expression and mapped according to the  
1081 table in the §5.

1082

### 1083 **5.2.6.2.1 RequestProcessingNotification message**

1084 The RequestProcessingNotification message format is defined as follows.

1085

```
1086 <RequestProcessingNotification CurrentTime="xs:dateTime" ...>
1087 <Request ...>
1088   <TransportInformation ...> {any}* </TransportInformation> ?
1089   <Message ...>
1090     <Size Unit=("bit" | "byte" | "word" | "dword" | "qword")
1091       ...>xs:positiveInteger</Size> ?
1092     (
1093       <NotIncluded/> |
1094       <Text>xs:string</Text> |
1095       <Binary>xs:base64Binary</Binary> |
1096       <Xml>{any}*</Xml>
1097     )
1098     {any}*
1099   </Message>
1100   {any}*
1101 </Request> ?
1102 <Reply ...>
1103   <!-- ... see contents of the Request element above ... -->
1104 </Reply> ?
1105 <StateInformation>
1106 <muws-xs2:StateTransition> <!-- ...see [MUWS]... --> </muws-xs2:StateTransition> +
1107 </StateInformation>
1108 {any}*
```

1109 </RequestProcessingNotification>

1110

1111 **RequestProcessingNotification** is a container element of the information about a request going  
1112 through the request processing states (Figure 12).

1113 **RequestProcessingNotification/@CurrentTime** indicates current time measured at the  
1114 manageability endpoint. All time/date values in this notification information are synchronized with  
1115 this time indication.

1116 **RequestProcessingNotification/Request** element contains information about the request itself.  
1117 Note that the request is not necessarily serialized as a SOAP message. Therefore, the contents  
1118 allow information about requests in general, however the information has to be serializable in  
1119 XML [XML]. The presence of this element in the notification MUST indicate presence of the  
1120 actual request message sent to the Web service endpoint being managed. The contents may  
1121 vary depending on what the implementation of the manageability endpoint can or intends to  
1122 provide. For example, for security reasons the actual contents of the message may be omitted.  
1123 However, in order to indicate that the request message exists, this element has to be included in  
1124 the notification.

1125 **RequestProcessingNotification/Request/TransportInformation** element contains information  
1126 about the transport by which the request was received. The content of this element is open, but  
1127 WSDM defines the following elements useable for TCP/IP transports.

```
1128 <TcpIpInfo
1129     Direction=("from" | "to")
1130     Port="xs:positiveInteger"
1131     Protocol=("TCP" | "UDP") ...>
1132     (
1133     <IPv4Address>
1134         xs:hexBinary[xs:length[@value="8" and @fixed="true"]]
1135     </IPv4Address> |
1136     <IPv6Address>
1137         xs:hexBinary[xs:length[@value="32" and @fixed="true"]]
1138     </IPv6Address>
1139     )
1140     {any}*
1141 </TcpIpInfo>
```

1142 **TcpIpInfo** contains information about a communication to or from an IP addressable  
1143 network device.

1144 **TcpIpInfo/@Direction** indicates communication to or from the IP addressable network  
1145 device.

1146 **TcpIpInfo/@Port** is a TCP/IP network port number used on the IP addressable network  
1147 device.

1148 **TcpIpInfo/@Protocol** indicates if the TCP or UDP protocol is used.

1149 **TcpIpInfo/IPv4Address** contains hexadecimal representation of the IP address version  
1150 4. The value MUST represent 32 bits.

1151 **TcpIpInfo/IPv6Address** contains hexadecimal representation of the IP address version  
1152 6. The value MUST represent 128 bits.

1153 **RequestProcessingNotification/Request/Message** element contains the message observed by  
1154 the Web service endpoint being managed.

1155 **RequestProcessingNotification/Request/Message/Size** indicates size of the message. When  
1156 subscribed to observations with attachments, this value includes the size of the message payload  
1157 plus all the attachments. Otherwise, just the payload of the message (i.e. size of the contents of  
1158 the RequestProcessingNotification/Request/Message element) is reported. Note that the actual  
1159 message contents may not be reported for security reasons, however size may be reported.

1160 **RequestProcessingNotification/Request/Message/Size/@Unit** indicates what units were used  
1161 to calculate the size of the message. The valid values of this attribute are:

1162 **bit** – size indicates number of bits in the message.

1163 **byte** – size indicates number of bytes (8 bit sets) in the message

1164 **word** – size indicates number of double bytes (16 bit sets) in the message.

1165 **dword** – size indicates number of double words (32 bit sets) in the message.

1166 **qword** – size indicates number of quad words (64 bit sets) in the message.

1167 **RequestProcessingNotification/Request/Message/NotIncluded** element indicates that the  
1168 message content is intentionally not provided by the implementation of the Web service endpoint  
1169 manageability.

1170 **RequestProcessingNotification/Request/Message/Text** element contains the observed  
1171 message's text representation. For example, a non-well formed XML message should be  
1172 represented as text. It is recommended that text data is wrapped in an XML CDATA section  
1173 **[XML]**.

1174 **RequestProcessingNotification/Request/Message/Binary** element contains the binary  
1175 representation of the observed message. If a message cannot be represented as either well-  
1176 formed XML nor as text, it should be binary encoded.

1177 **RequestProcessingNotification/Request/Message/XML** element contains the observed  
1178 message's XML representation. For example, a SOAP message envelope element (S:Envelope)  
1179 may appear in the contents.

1180 **RequestProcessingNotification/Request/{any}** is an extensibility element where additional  
1181 information about the request MAY appear. The form of the information representation in XML is  
1182 manageability endpoint implementation specific. In other words, vendor extensions may appear  
1183 here.

1184 The **RequestProcessingNotification/Reply** element contains information about the reply (if any)  
1185 for the request. Note that fault is also a valid reply element. The content of this element has the  
1186 same format as the RequestProcessingNotification/Request element.

1187 **RequestProcessingNotification/StateInformation** element contains information about the  
1188 request processing state.

1189 **RequestProcessingNotification/StateInformation/muws-xs2:StateTransition** element  
1190 contains information about a state transition. There MUST be exactly one such element for each  
1191 state change event. There MUST be one or more such elements for the digest event.

1192 **RequestProcessingNotification/StateInformation/muws-xs2:StateTransition/@muws-**  
1193 **xs2:Time** indicates time when the described transition occurred. Note that according to the  
1194 request processing state model (Figure 12), all transitions are instantaneous. Time is measured  
1195 at the implementation of the manageability endpoint and is synchronized with the  
1196 RequestProcessingNotification/@CurrentTime value reading.

1197 **RequestProcessingNotification/StateInformation/muws-xs2:StateTransition/muws-**  
1198 **xs2:EnteredState** indicates which request processing state was entered.

1199 **RequestProcessingNotification/StateInformation/muws-xs2:StateTransition/muws-**  
1200 **xs2:PreviousState** indicates which request processing state was exited.

1201 **RequestProcessingNotification/{any}** is an extensibility element where additional information  
1202 about this request processing notification MAY appear. The form of the information representation  
1203 in XML is manageability endpoint implementation specific. In other words, vendor extensions may  
1204 appear here.

1205 The contents of the RequestProcessingNotification element SHOULD be used to specify  
1206 selectors **[WS-N]** when subscribing to notification messages containing this element.



1207 **5.2.6.2.2 Examples of events against the Web service endpoint request**  
1208 **processing state**

1209 Consider the following message exchange with a fictitious order-entry Web service endpoint.

1210

1211 Request:

```
1212 <S:Envelope xmlns:x="..." ... >  
1213 . . .  
1214 <S:Body>  
1215     <x:Order>  
1216         <x:Item>...</x:Item>  
1217         <x:Quantity>...</x:Quantity>  
1218     </x:Order>  
1219 </S:Body>  
1220 </S:Envelope>
```

1221

1222 Reply:

```
1223 <S:Envelope xmlns:x="..." ... >  
1224 . . .  
1225 <S:Body>  
1226     <x:Shipped>  
1227         <x:Item>...</x:Item>  
1228         <x:Quantity>...</x:Quantity>  
1229     </x:Shipped>  
1230 </S:Body>  
1231 </S:Envelope>
```

1232

1233 To be notified of a particular item shortage when the order request is processed and the shipped  
1234 quantity is less than the ordered quantity, the following XPath selector should be specified when  
1235 subscribing to the **mows-events:ManageableEndpoint/mows-**  
1236 **events:RequestProcessingObservations/mows-events:RequestCompleted** topic.

1237

1238 Selector:

```
1239 boolean(//mows-xs:RequestProcessingNotification[mows-xs:Request/mows-xs:Message/mows-  
1240 xs:Xml//x:Order/x:Quantity < mows-xs:Reply/mows-xs:Message/mows-  
1241 xs:Xml//x:Shipped/x:Quantity])
```

1242

1243 This way, when the condition is met, the manageable Web service endpoint will emit the  
1244 notification message containing the **RequestProcessingNotification** element with the following  
1245 contents.

1246

```
1247 <RequestProcessingNotification CurrentTime="...">  
1248 <Request>  
1249     <TransportInformation>  
1250         <TcpIpInfo Direction="from" Port="2840" Protocol="TCP">  
1251             <IPV4Address>C0A80002</IPV4Address>  
1252         </TcpIpInfo>  
1253         <TcpIpInfo Direction="to" Port="80" Protocol="TCP">  
1254             <IPV4Address>C0A80003</IPV4Address>  
1255         </TcpIpInfo>  
1256     </TransportInformation>
```

```

1257     <Message>
1258     <Size Unit="byte">257</Size>
1259     <Xml>
1260     <S:Envelope xmlns:S="..." xmlns:x="..." ...>
1261     . . .
1262     <S:Body>
1263         <x:Order>
1264             <x:Item>123</x:Item>
1265             <x:Quantity>10</x:Quantity>
1266         </x:Order>
1267     </S:Body>
1268 </S:Envelope>
1269 </Xml>
1270 </Message>
1271 </Request>
1272 <Reply>
1273     <TransportInformation>
1274         <TcpIpInfo Direction="to" Port="2840" Protocol="TCP">
1275             <IPv4Address>C0A80002</IPv4Address>
1276         </TcpIpInfo>
1277         <TcpIpInfo Direction="from" Port="80" Protocol="TCP">
1278             <IPv4Address>C0A80003</IPv4Address>
1279         </TcpIpInfo>
1280     </TransportInformation>
1281     <Message>
1282     <Size Unit="byte">232</Size>
1283     <Xml>
1284     <S:Envelope xmlns:S="..." xmlns:x="..." ...>
1285     . . .
1286     <S:Body>
1287         <x:Shipped>
1288             <x:Item>123</x:Item>
1289             <x:Quantity>2</x:Quantity>
1290         </x:Shipped>
1291     </S:Body>
1292 </S:Envelope>
1293 </Xml>
1294 </Message>
1295 </Reply>
1296 <muws-xs2:StateTransition Time="...">
1297 <muws-xs2:EnteredState/><RequestCompletedState/></muws-xs2:EnteredState>
1298 <muws-xs2:PreviousState><RequestProcessingState/></muws-xs2:PreviousState>
1299 </muws-xs2:StateTransition>
1300 . . .
1301 </RequestProcessingNotification>
1302
1303

```

1304

## 6 References

1305

### 6.1 Normative

- 1306 [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,  
1307 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 1308 [MUWS] William Vambenepe, *Web Services Distributed*  
1309 *Management:Management using Web Services (MUWS 1.0) Part 1*,  
1310 OASIS Committee Draft, December 2004, [http://docs.oasis-](http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part1-1.0.pdf)  
1311 [open.org/wsdm/2004/12/wsdm-muws-part1-1.0.pdf](http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part1-1.0.pdf)  
1312 William Vambenepe, *Web Services Distributed*  
1313 *Management:Management using Web Services (MUWS 1.0) Part 2*,  
1314 OASIS Committee Draft, December 2004, [http://docs.oasis-](http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part2-1.0.pdf)  
1315 [open.org/wsdm/2004/12/wsdm-muws-part2-1.0.pdf](http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part2-1.0.pdf)
- 1316 [WS-A] Don Box, et al., *Web services Addressing (WS-Addressing)*, W3C  
1317 Member Submission, August 2004,  
1318 <http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/>
- 1319 [WS-RP] Steve Graham, et al., *Web Services Resource Properties 1.2 (WS-*  
1320 *ResourceProperties)*, OASIS Working Draft, June 2004,  
1321 [http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-](http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-1.2-draft-04.pdf)  
1322 [1.2-draft-04.pdf](http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-1.2-draft-04.pdf)
- 1323 [WS-N] Steve Graham, et al., *Web Services Base Notification 1.2 (WS-*  
1324 *BaseNotification)*, OASIS Working Draft, June 2004, [http://docs.oasis-](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf)  
1325 [open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf)
- 1326 [WS-T] William Vambenepe, *Web Services Topics 1.2 (WS-Topics)*, OASIS  
1327 Working Draft, July 2004, [http://docs.oasis-open.org/wsn/2004/06/wsn-](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.pdf)  
1328 [WS-Topics-1.2-draft-01.pdf](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.pdf)
- 1329 [WSDL] Erik Christensen, et al., *Web services Description Language (WSDL)*  
1330 *1.1*, W3C Note, March 2001, <http://www.w3.org/TR/wSDL>
- 1331 [SOAP] Don Box, et al., *Simple Object Access Protocol 1.1*, W3C Note, May  
1332 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- 1333 [XMLS] Henry S. Thompson, et al. *XML Schema Part 1: Structures*, W3C  
1334 Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-1/>  
1335 Paul V. Biron, et al. *XML Schema Part 2: Datatypes*, W3C  
1336 Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-2/>
- 1337 [XML] Tim Bray, et al., *Extensible Markup Language (XML) 1.0 (Third Edition)*,  
1338 W3C Recommendation, February 2004, <http://www.w3.org/TR/REC-xml>
- 1339 [XNS] Tim Bray, et al., *Namespaces in XML*, W3C Recommendation, January  
1340 1999, <http://www.w3.org/TR/REC-xml-names/>
- 1341 [XPath] James Clark, et al., *XML Path Language (XPath) Version 1.0*, W3C  
1342 Recommendation, November 1999, [http://www.w3.org/TR/1999/REC-](http://www.w3.org/TR/1999/REC-xpath-19991116)  
1343 [xpath-19991116](http://www.w3.org/TR/1999/REC-xpath-19991116)

1344

### 6.2 Non-normative

- 1345 [MOWS-Reqs] Mark Potts, et al., *WSDM Management of Web Services Requirements*,  
1346 October 2003, [http://www.oasis-](http://www.oasis-open.org/apps/org/workgroup/wsdm/download.php/3887/WSDM-MOWS-Requirements.20031008.doc)  
1347 [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-Requirements.20031008.doc)  
1348 [Requirements.20031008.doc](http://www.oasis-open.org/apps/org/workgroup/wsdm/download.php/3887/WSDM-MOWS-Requirements.20031008.doc)

1349	<b>[WS-Arch]</b>	David Booth, et al. <i>Web Services Architecture</i> , W3C Working Group Note, February 2004, <a href="http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/">http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/</a>
1350		
1351		
1352	<b>[WSLC]</b>	Hao He, et al., <i>Web Service Management: Service Lifecycle</i> , W3C Note, February 2004, <a href="http://www.w3.org/TR/2004/NOTE-wslc-20040211/">http://www.w3.org/TR/2004/NOTE-wslc-20040211/</a>
1353		
1354	<b>[WSS]</b>	Anthony Nadalin, et al., <i>Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)</i> , March 2004, OASIS Standard, <a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf</a>
1355		
1356		
1357	<b>[BSP]</b>	Abbie Barbir, et al., <i>Basic Security Profile Version 1.0</i> , WS-I Working Group Draft, May 2004, <a href="http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0-2004-05-12.html">http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0-2004-05-12.html</a>
1358		
1359		
1360		

---

1361

## Appendix A. Acknowledgments

1362 The following people made contributions to this specification: Brian Carol, Fred Carter, John  
1363 DeCarlo, Andreas Dharmawan, Maryann Hondo, Heather Kreger, Bryan Murray, Micheal Perks,  
1364 Igor Sedukhin, William Vambenepe, Andrea Westerinen.

1365

1366 The following individuals were members of the committee when the specification was approved  
1367 by the technical committee: Guru Bhat, Jeff Bohren, Winston Bumpus, Nick Butler, Brian Carroll,  
1368 Fred Carter, Michael Clements, David Cox, John DeCarlo, Andreas Dharmawan, Mark Ellison,  
1369 John Fuller, Paul Lipton, Heather Kreger, Hal Lockhart, Frederico Maciel, Tom Maguire, Bryan  
1370 Murray, Richard Nikula, Mark Peel, Richard Pelavin, Hodayoun Pourheidari, Warren Roberts,  
1371 Karl Schopmeyer, Igor Sedukhin, David Snelling, Thomas Studwell, William Vambenepe, Andrea  
1372 Westerinen, Jim Willits, Zhili Zhang.

---

## Appendix B. Revision History

Rev	Date	By Whom
wd	2003-10-31	Igor Sedukhin
wd	2003-11-14	Igor Sedukhin
wd	2003-12-02	Igor Sedukhin
wd	2004-01-26	Igor Sedukhin
wd	2004-02-17	Igor Sedukhin
wd	2004-03-01	Igor Sedukhin
wd	2004-03-18	Igor Sedukhin
wd	2004-03-19	Igor Sedukhin
wd	2004-03-24	Igor Sedukhin
wd	2004-03-24	Igor Sedukhin
cd	2004-04-02	Igor Sedukhin
wd	2004-07-21	Igor Sedukhin
wd	2004-09-11	Igor Sedukhin
wd	2004-10-11	Igor Sedukhin
wd	2004-10-24	Igor Sedukhin
wd	2004-11-04	Igor Sedukhin
wd	2004-11-15	Igor Sedukhin
wd	2004-11-19	Igor Sedukhin
wd	2004-11-23	Igor Sedukhin
wd	2004-12-03	Igor Sedukhin
cd	2004-12-10	Igor Sedukhin
standard	2005-03-09	Igor Sedukhin

---

## Appendix C. Notices

1375

1376 OASIS takes no position regarding the validity or scope of any intellectual property or other rights  
1377 that might be claimed to pertain to the implementation or use of the technology described in this  
1378 document or the extent to which any license under such rights might or might not be available;  
1379 neither does it represent that it has made any effort to identify any such rights. Information on  
1380 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS  
1381 website. Copies of claims of rights made available for publication and any assurances of licenses  
1382 to be made available, or the result of an attempt made to obtain a general license or permission  
1383 for the use of such proprietary rights by implementors or users of this specification, can be  
1384 obtained from the OASIS Executive Director.

1385 OASIS invites any interested party to bring to its attention any copyrights, patents or patent  
1386 applications, or other proprietary rights which may cover technology that may be required to  
1387 implement this specification. Please address the information to the OASIS Executive Director.

1388 Copyright © OASIS Open 2003-2004. *All Rights Reserved.*

1389 This document and translations of it may be copied and furnished to others, and derivative works  
1390 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,  
1391 published and distributed, in whole or in part, without restriction of any kind, provided that the  
1392 above copyright notice and this paragraph are included on all such copies and derivative works.  
1393 However, this document itself does not be modified in any way, such as by removing the  
1394 copyright notice or references to OASIS, except as needed for the purpose of developing OASIS  
1395 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual  
1396 Property Rights document must be followed, or as required to translate it into languages other  
1397 than English.

1398 The limited permissions granted above are perpetual and will not be revoked by OASIS or its  
1399 successors or assigns.

1400 This document and the information contained herein is provided on an "AS IS" basis and OASIS  
1401 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO  
1402 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE  
1403 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A  
1404 PARTICULAR PURPOSE.

1405

## Appendix D. XML Schemas

```

1407 <?xml version="1.0" encoding="utf-8"?>
1408 <xs:schema targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-
1409 mows.xsd" xmlns:mows-xs="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.xsd"
1410 xmlns:muws-xs2="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part2.xsd"
1411 xmlns:muws-xs1="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part1.xsd"
1412 xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1413 xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
1414 attributeFormDefault="unqualified">
1415
1416     <xs:import namespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1417 part1.xsd" schemaLocation="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1418 part1.xsd"/>
1419     <xs:import namespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1420 part2.xsd" schemaLocation="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1421 part2.xsd"/>
1422     <xs:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
1423 schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
1424
1425     <!-- MOWS::ManageabilityReferences -->
1426     <xs:element name="GetManageabilityReferences"/>
1427     <xs:element name="GetManageabilityReferencesResponse">
1428         <xs:complexType>
1429             <xs:sequence>
1430                 <xs:element ref="muws-xs1:ManageabilityEndpointReference"
1431                     maxOccurs="unbounded"/>
1432             </xs:sequence>
1433         </xs:complexType>
1434     </xs:element>
1435
1436     <!-- MOWS::Identification -->
1437     <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
1438     <xs:element name="EndpointDescriptions">
1439         <xs:complexType>
1440             <xs:sequence>
1441                 <xs:element name="description" type="xs:anyURI"
1442                     minOccurs="0" maxOccurs="unbounded"/>
1443             </xs:sequence>
1444             <xs:anyAttribute namespace="##other" processContents="lax"/>
1445         </xs:complexType>
1446     </xs:element>
1447
1448     <xs:complexType name="EndpointIdentificationPropertiesType">
1449         <xs:sequence>
1450             <xs:element ref="mows-xs:EndpointReference"/>
1451             <xs:element ref="mows-xs:EndpointDescriptions" minOccurs="0"/>
1452         </xs:sequence>
1453     </xs:complexType>
1454     <xs:element name="EndpointIdentificationProperties"
1455         type="mows-xs:EndpointIdentificationPropertiesType"/>
1456
1457     <!-- MOWS::Metrics -->

```



```

1458 <xs:complexType name="IntegerCounter">
1459     <xs:simpleContent>
1460         <xs:extension base="xs:nonNegativeInteger">
1461             <xs:attributeGroup ref="muws-xs2:MetricAttributes"/>
1462             <xs:anyAttribute namespace="##other" processContents="lax"/>
1463         </xs:extension>
1464     </xs:simpleContent>
1465 </xs:complexType>
1466
1467 <xs:complexType name="DurationMetric">
1468     <xs:simpleContent>
1469         <xs:extension base="xs:duration">
1470             <xs:attributeGroup ref="muws-xs2:MetricAttributes"/>
1471             <xs:anyAttribute namespace="##other" processContents="lax"/>
1472         </xs:extension>
1473     </xs:simpleContent>
1474 </xs:complexType>
1475
1476 <xs:element name="NumberOfRequests" type="mows-xs:IntegerCounter"/>
1477 <xs:element name="NumberOfSuccessfulRequests" type="mows-xs:IntegerCounter"/>
1478 <xs:element name="NumberOfFailedRequests" type="mows-xs:IntegerCounter"/>
1479 <xs:element name="ServiceTime" type="mows-xs:DurationMetric"/>
1480 <xs:element name="MaxResponseTime" type="mows-xs:DurationMetric"/>
1481 <xs:element name="LastResponseTime" type="mows-xs:DurationMetric"/>
1482
1483 <xs:complexType name="EndpointMetricsPropertiesType">
1484     <xs:sequence>
1485         <xs:element ref="mows-xs:NumberOfRequests" minOccurs="0"/>
1486         <xs:element ref="mows-xs:NumberOfFailedRequests" minOccurs="0"/>
1487         <xs:element ref="mows-xs:NumberOfSuccessfulRequests"
1488             minOccurs="0"/>
1489         <xs:element ref="mows-xs:ServiceTime" minOccurs="0"/>
1490         <xs:element ref="mows-xs:MaxResponseTime" minOccurs="0"/>
1491         <xs:element ref="mows-xs:LastResponseTime" minOccurs="0"/>
1492     </xs:sequence>
1493 </xs:complexType>
1494 <xs:element name="EndpointMetricsProperties"
1495     type="mows-xs:EndpointMetricsPropertiesType"/>
1496
1497 <!-- MOWS::OperationalState -->
1498 <xs:complexType name="OperationalStateType">
1499     <xs:complexContent>
1500         <xs:extension base="muws-xs2:StateType"/>
1501     </xs:complexContent>
1502 </xs:complexType>
1503 <xs:element name="UpState">
1504     <xs:complexType>
1505         <xs:complexContent>
1506             <xs:restriction base="mows-xs:OperationalStateType"/>
1507         </xs:complexContent>
1508     </xs:complexType>
1509 </xs:element>
1510 <xs:element name="IdleState">
1511     <xs:complexType>
1512         <xs:complexContent>
1513             <xs:restriction base="mows-xs:OperationalStateType">

```

```

1514         <xs:sequence>
1515             <xs:element ref="mows-xs:UpState"/>
1516         </xs:sequence>
1517     </xs:restriction>
1518 </xs:complexContent>
1519 </xs:complexType>
1520 </xs:element>
1521 <xs:element name="BusyState">
1522     <xs:complexType>
1523         <xs:complexContent>
1524             <xs:restriction base="mows-xs:OperationalStateType">
1525                 <xs:sequence>
1526                     <xs:element ref="mows-xs:UpState"/>
1527                 </xs:sequence>
1528             </xs:restriction>
1529         </xs:complexContent>
1530     </xs:complexType>
1531 </xs:element>
1532 <xs:element name="DownState">
1533     <xs:complexType>
1534         <xs:complexContent>
1535             <xs:restriction base="mows-xs:OperationalStateType"/>
1536         </xs:complexContent>
1537     </xs:complexType>
1538 </xs:element>
1539 <xs:element name="StoppedState">
1540     <xs:complexType>
1541         <xs:complexContent>
1542             <xs:restriction base="mows-xs:OperationalStateType">
1543                 <xs:sequence>
1544                     <xs:element ref="mows-xs:DownState"/>
1545                 </xs:sequence>
1546             </xs:restriction>
1547         </xs:complexContent>
1548     </xs:complexType>
1549 </xs:element>
1550 <xs:element name="CrashedState">
1551     <xs:complexType>
1552         <xs:complexContent>
1553             <xs:restriction base="mows-xs:OperationalStateType">
1554                 <xs:sequence>
1555                     <xs:element ref="mows-xs:DownState"/>
1556                 </xs:sequence>
1557             </xs:restriction>
1558         </xs:complexContent>
1559     </xs:complexType>
1560 </xs:element>
1561 <xs:element name="SaturatedState">
1562     <xs:complexType>
1563         <xs:complexContent>
1564             <xs:restriction base="mows-xs:OperationalStateType">
1565                 <xs:sequence>
1566                     <xs:element ref="mows-xs:DownState"/>
1567                 </xs:sequence>
1568             </xs:restriction>
1569         </xs:complexContent>

```

```

1570         </xs:complexType>
1571     </xs:element>
1572
1573     <xs:element name="CurrentOperationalState" type="mows-xs:OperationalStateType"/>
1574     <xs:element name="LastOperationalStateTransition"
1575         type="muws-xs2:StateTransitionType"/>
1576
1577     <xs:complexType name="EndpointOperationalStatePropertiesType">
1578         <xs:sequence>
1579             <xs:element ref="mows-xs:CurrentOperationalState"/>
1580             <xs:element ref="mows-xs:LastOperationalStateTransition"
1581                 minOccurs="0"/>
1582         </xs:sequence>
1583     </xs:complexType>
1584     <xs:element name="EndpointOperationalStateProperties"
1585         type="mows-xs:EndpointOperationalStatePropertiesType"/>
1586
1587     <!-- MOWS::RequestProcessingState -->
1588     <xs:complexType name="RequestProcessingStateType">
1589         <xs:complexContent>
1590             <xs:extension base="muws-xs2:StateType"/>
1591         </xs:complexContent>
1592     </xs:complexType>
1593     <xs:element name="RequestReceivedState">
1594         <xs:complexType>
1595             <xs:complexContent>
1596                 <xs:restriction base="mows-xs:RequestProcessingStateType"/>
1597             </xs:complexContent>
1598         </xs:complexType>
1599     </xs:element>
1600     <xs:element name="RequestProcessingState">
1601         <xs:complexType>
1602             <xs:complexContent>
1603                 <xs:restriction base="mows-xs:RequestProcessingStateType"/>
1604             </xs:complexContent>
1605         </xs:complexType>
1606     </xs:element>
1607     <xs:element name="RequestCompletedState">
1608         <xs:complexType>
1609             <xs:complexContent>
1610                 <xs:restriction base="mows-xs:RequestProcessingStateType"/>
1611             </xs:complexContent>
1612         </xs:complexType>
1613     </xs:element>
1614     <xs:element name="RequestFailedState">
1615         <xs:complexType>
1616             <xs:complexContent>
1617                 <xs:restriction base="mows-xs:RequestProcessingStateType"/>
1618             </xs:complexContent>
1619         </xs:complexType>
1620     </xs:element>
1621     <xs:complexType name="MessageContentNotIncludedFlag"/>
1622     <xs:simpleType name="MessageSizeUnitType">
1623         <xs:restriction base="xs:string">
1624             <xs:enumeration value="bit"/>
1625             <xs:enumeration value="byte"/>

```

```

1626         <xs:enumeration value="word"/>
1627         <xs:enumeration value="dword"/>
1628         <xs:enumeration value="qword"/>
1629     </xs:restriction>
1630 </xs:simpleType>
1631 <xs:complexType name="MessageContentSizeType">
1632     <xs:simpleContent>
1633         <xs:extension base="xs:positiveInteger">
1634             <xs:attribute name="Unit"
1635                 type="mows-xs:MessageSizeUnitType" use="required"/>
1636             <xs:anyAttribute namespace="##other" processContents="lax"/>
1637         </xs:extension>
1638     </xs:simpleContent>
1639 </xs:complexType>
1640 <xs:complexType name="MessageContentType">
1641     <xs:sequence>
1642         <xs:element name="Size"
1643             type="mows-xs:MessageContentSizeType" minOccurs="0"/>
1644         <xs:choice>
1645             <xs:element name="NotIncluded"
1646                 type="mows-xs:MessageContentNotIncludedFlag"/>
1647             <xs:element name="Text" type="xs:string"/>
1648             <xs:element name="Binary" type="xs:base64Binary"/>
1649             <xs:element name="Xml"
1650                 type="mows-xs:AnyXmlContentsType"/>
1651         </xs:choice>
1652         <xs:any namespace="##other" processContents="lax"
1653             minOccurs="0" maxOccurs="unbounded"/>
1654     </xs:sequence>
1655     <xs:anyAttribute namespace="##other" processContents="lax"/>
1656 </xs:complexType>
1657 <xs:complexType name="AnyXmlContentsType">
1658     <xs:sequence>
1659         <xs:any namespace="##any" processContents="lax"
1660             minOccurs="0" maxOccurs="unbounded"/>
1661     </xs:sequence>
1662     <xs:anyAttribute namespace="##any" processContents="lax"/>
1663 </xs:complexType>
1664 <xs:complexType name="MessageInformationType">
1665     <xs:sequence>
1666         <xs:element name="TransportInformation"
1667             type="mows-xs:AnyXmlContentsType" minOccurs="0"/>
1668         <xs:element name="Message" type="mows-xs:MessageContentType"/>
1669         <xs:any namespace="##any" processContents="lax"
1670             minOccurs="0" maxOccurs="unbounded"/>
1671     </xs:sequence>
1672     <xs:anyAttribute namespace="##any" processContents="lax"/>
1673 </xs:complexType>
1674 <xs:complexType name="RequestProcessingStateInformationType">
1675     <xs:sequence>
1676         <xs:element ref="muws-xs2:StateTransition" maxOccurs="unbounded"/>
1677     </xs:sequence>
1678 </xs:complexType>
1679 <xs:element name="RequestProcessingNotification">
1680     <xs:complexType>
1681         <xs:sequence>

```

```

1682         <xs:element name="Request"
1683             type="mows-xs:MessageInformationType"
1684             minOccurs="0"/>
1685         <xs:element name="Reply"
1686             type="mows-xs:MessageInformationType"
1687             minOccurs="0"/>
1688         <xs:element name="StateInformation"
1689             type="mows-
1690 xs:RequestProcessingStateInformationType"/>
1691         <xs:any namespace="##any" processContents="lax"
1692             minOccurs="0" maxOccurs="unbounded"/>
1693     </xs:sequence>
1694     <xs:attribute name="CurrentTime" type="xs:dateTime" use="required"/>
1695     <xs:anyAttribute namespace="##any" processContents="lax"/>
1696 </xs:complexType>
1697 </xs:element>
1698 <xs:simpleType name="IPv4AddressType">
1699     <xs:restriction base="xs:hexBinary">
1700         <xs:length value="8" fixed="true"/>
1701     </xs:restriction>
1702 </xs:simpleType>
1703 <xs:element name="IPv4Address" type="mows-xs:IPv4AddressType"/>
1704 <xs:simpleType name="IPv6AddressType">
1705     <xs:restriction base="xs:hexBinary">
1706         <xs:length value="32" fixed="true"/>
1707     </xs:restriction>
1708 </xs:simpleType>
1709 <xs:element name="IPv6Address" type="mows-xs:IPv6AddressType"/>
1710 <xs:simpleType name="TcplpDirectionType">
1711     <xs:restriction base="xs:string">
1712         <xs:enumeration value="to"/>
1713         <xs:enumeration value="from"/>
1714     </xs:restriction>
1715 </xs:simpleType>
1716 <xs:simpleType name="TcplpProtocolType">
1717     <xs:restriction base="xs:string">
1718         <xs:enumeration value="TCP"/>
1719         <xs:enumeration value="UDP"/>
1720     </xs:restriction>
1721 </xs:simpleType>
1722 <xs:element name="TcplpInfo">
1723     <xs:complexType>
1724         <xs:sequence>
1725             <xs:choice>
1726                 <xs:element ref="mows-xs:IPv4Address"/>
1727                 <xs:element ref="mows-xs:IPv6Address"/>
1728             </xs:choice>
1729             <xs:any namespace="##any" processContents="lax"
1730                 minOccurs="0" maxOccurs="unbounded"/>
1731         </xs:sequence>
1732         <xs:attribute name="Direction" type="mows-xs:TcplpDirectionType"
1733             use="required"/>
1734         <xs:attribute name="Port" type="xs:positiveInteger" use="required"/>
1735         <xs:attribute name="Protocol" type="mows-xs:TcplpProtocolType"
1736             use="required"/>
1737         <xs:anyAttribute namespace="##any" processContents="lax"/>

```

```
1738         </xs:complexType>
1739     </xs:element>
1740 </xs:schema>
1741
```

## Appendix E. WSDL elements

```
1743 <?xml version="1.0" encoding="utf-8"?>
1744 <definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
1745 xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:wsrf-rp="http://docs.oasis-
1746 open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.xsd" xmlns:mows-
1747 xs="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.xsd" xmlns:mows-
1748 wsdl="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.wsdl"
1749 targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.wsdl">
1750
1751     <types>
1752         <xs:schema elementFormDefault="qualified"
1753 targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.wsdl">
1754             <xs:import namespace="http://docs.oasis-
1755 open.org/wsdm/2004/12/mows/wsdm-mows.xsd" schemaLocation="http://docs.oasis-
1756 open.org/wsdm/2004/12/mows/wsdm-mows.xsd"/>
1757             </xs:schema>
1758         </types>
1759
1760         <message name="GetManageabilityReferencesRequest">
1761             <part name="body" element="mows-xs:GetManageabilityReferences"/>
1762         </message>
1763         <message name="GetManageabilityReferencesResponse">
1764             <part name="body"
1765                 element="mows-xs:GetManageabilityReferencesResponse"/>
1766         </message>
1767
1768         <portType name="ManageabilityReferences">
1769             <operation name="GetManageabilityReferences">
1770                 <input name="GetManageabilityReferencesRequest"
1771                     message="mows-wsdl:GetManageabilityReferencesRequest"/>
1772                 <output name="GetManageabilityReferencesResponse"
1773                     message="mows-wsdl:GetManageabilityReferencesResponse"/>
1774             </operation>
1775         </portType>
1776
1777         <portType name="EndpointIdentification"
1778             wsrf-rp:ResourceProperties="mows-xs:EndpointIdentificationProperties"/>
1779
1780         <portType name="EndpointMetrics"
1781             wsrf-rp:ResourceProperties="mows-xs:EndpointMetricsProperties"/>
1782
1783         <portType name="EndpointOperationalState"
1784             wsrf-rp:ResourceProperties="mows-xs:EndpointOperationalStateProperties"/>
1785     </definitions>
1786
```

## Appendix F. Notification topic spaces

```

1788 <?xml version="1.0" encoding="utf-8"?>
1789 <wstop:TopicSpace name="MOWS"
1790     targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows-
1791 events.xml"
1792     xmlns:muws-xs2="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1793 part2.xsd"
1794     xmlns:muws-xs1="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
1795 part1.xsd"
1796     xmlns:mows-xs="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.xsd"
1797     xmlns:wstop="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.xsd">
1798
1799 <wstop:Topic name="IdentificationCapability" messageTypes="muws-xs1:ManagementEvent"/>
1800 <wstop:Topic name="MetricsCapability" messageTypes="muws-xs1:ManagementEvent"/>
1801 <wstop:Topic name="OperationalStateCapability" messageTypes="muws-
1802 xs1:ManagementEvent"/>
1803 <wstop:Topic name="OperationalStatusCapability" messageTypes="muws-
1804 xs1:ManagementEvent"/>
1805 <wstop:Topic name="RequestProcessingStateCapability" messageTypes="muws-
1806 xs1:ManagementEvent"/>
1807
1808
1809 <wstop:Topic name="RequestProcessingObservations"
1810     messageTypes="muws-xs1:ManagementEvent">
1811     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1812 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1813 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1814 xs:RequestProcessingNotification)=1]
1815     </wstop:MessagePattern>
1816     <wstop:Topic name="RequestReceived"
1817         messageTypes="muws-xs1:ManagementEvent">
1818         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1819 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1820 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1821 xs:RequestProcessingNotification)=1]
1822         </wstop:MessagePattern>
1823     </wstop:Topic>
1824     <wstop:Topic name="RequestProcessing"
1825         messageTypes="muws-xs1:ManagementEvent">
1826         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1827 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1828 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1829 xs:RequestProcessingNotification)=1]
1830         </wstop:MessagePattern>
1831     </wstop:Topic>
1832     <wstop:Topic name="RequestCompleted"
1833         messageTypes="muws-xs1:ManagementEvent">
1834         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1835 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1836 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1837 xs:RequestProcessingNotification)=1]
1838     </wstop:MessagePattern>

```



```

1839     </wstop:Topic>
1840     <wstop:Topic name="RequestFailed"
1841         messageTypes="muws-xs1:ManagementEvent">
1842         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1843 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1844 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1845 xs:RequestProcessingNotification)=1]
1846         </wstop:MessagePattern>
1847         </wstop:Topic>
1848     <wstop:Topic name="Digest"
1849         messageTypes="muws-xs1:ManagementEvent">
1850         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1851 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1852 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1853 xs:RequestProcessingNotification)=1]
1854         </wstop:MessagePattern>
1855     </wstop:Topic>
1856 </wstop:Topic>
1857
1858 <wstop:Topic name="RequestProcessingObservationsWithAttachments"
1859     messageTypes="muws-xs1:ManagementEvent">
1860     <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1861 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1862 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1863 xs:RequestProcessingNotification)=1]
1864     </wstop:MessagePattern>
1865     <wstop:Topic name="RequestReceived"
1866         messageTypes="muws-xs1:ManagementEvent">
1867         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1868 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1869 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1870 xs:RequestProcessingNotification)=1]
1871         </wstop:MessagePattern>
1872         </wstop:Topic>
1873     <wstop:Topic name="RequestProcessing"
1874         messageTypes="muws-xs1:ManagementEvent">
1875         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1876 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1877 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1878 xs:RequestProcessingNotification)=1]
1879         </wstop:MessagePattern>
1880     </wstop:Topic>
1881     <wstop:Topic name="RequestCompleted"
1882         messageTypes="muws-xs1:ManagementEvent">
1883         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1884 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1885 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1886 xs:RequestProcessingNotification)=1]
1887         </wstop:MessagePattern>
1888     </wstop:Topic>
1889     <wstop:Topic name="RequestFailed"
1890         messageTypes="muws-xs1:ManagementEvent">
1891         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1892 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1893 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1894 xs:RequestProcessingNotification)=1]

```

```
1895     </wstop:MessagePattern>
1896     </wstop:Topic>
1897     <wstop:Topic name="Digest"
1898         messageTypes="muws-xs1:ManagementEvent">
1899         <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
1900 //muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
1901 xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
1902 xs:RequestProcessingNotification)=1]
1903         </wstop:MessagePattern>
1904     </wstop:Topic>
1905 </wstop:Topic>
1906 </wstop:TopicSpace>
```