



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

OASIS-Committee Draft, February 23, 2006

## Document identifier:

wsdm-mows-1.1-spec-cd-01

## Location:

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

## Editors:

*Kirk Wilson, Computer Associates* [kirk.wilson@ca.com](mailto:kirk.wilson@ca.com)

*Igor Sedukhin, Computer Associates.*

## Abstract:

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

## Status:

This document is an OASIS Working Draft.

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.

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

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

The errata document for this specification is maintained at:

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

## Table of Contents

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

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

---

# 1 Introduction

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

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

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

- WS Architecture [WS-Arch]
- XML [XML]
- XML Namespaces [XNS]
- XML Schema [XMLS]
- SOAP [SOAP]
- WSDL [WSDL]
- WS-Addressing [WS-A]
- WS-ResourceProperties [WS-RP]
- WS-BaseNotification [WS-N]
- WS-Topics [WS-T]
- XML Path Language [XPath]

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

## 1.1 Terminology

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

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

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

## 1.2 Notational conventions

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

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

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

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

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

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

## 2 Architecture

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

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

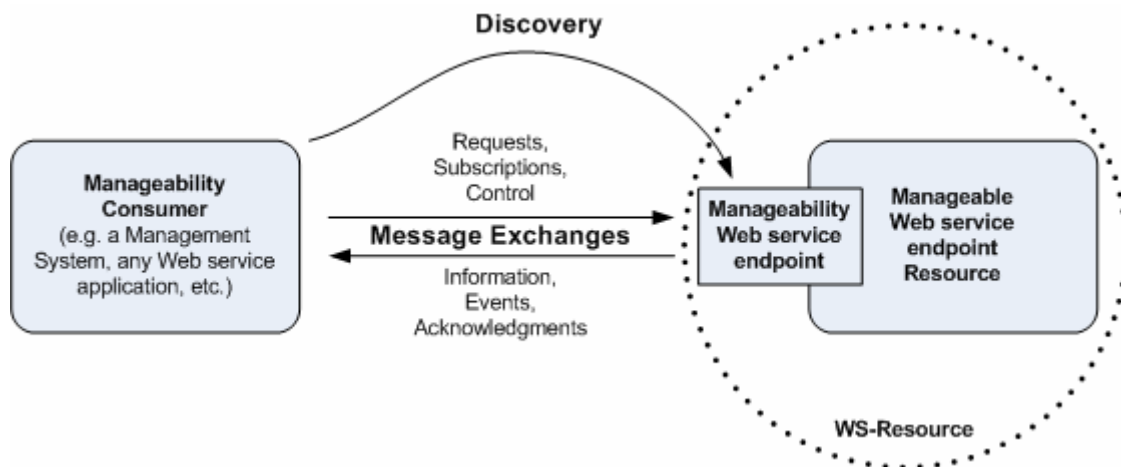


Figure 1. Management of Web services concepts

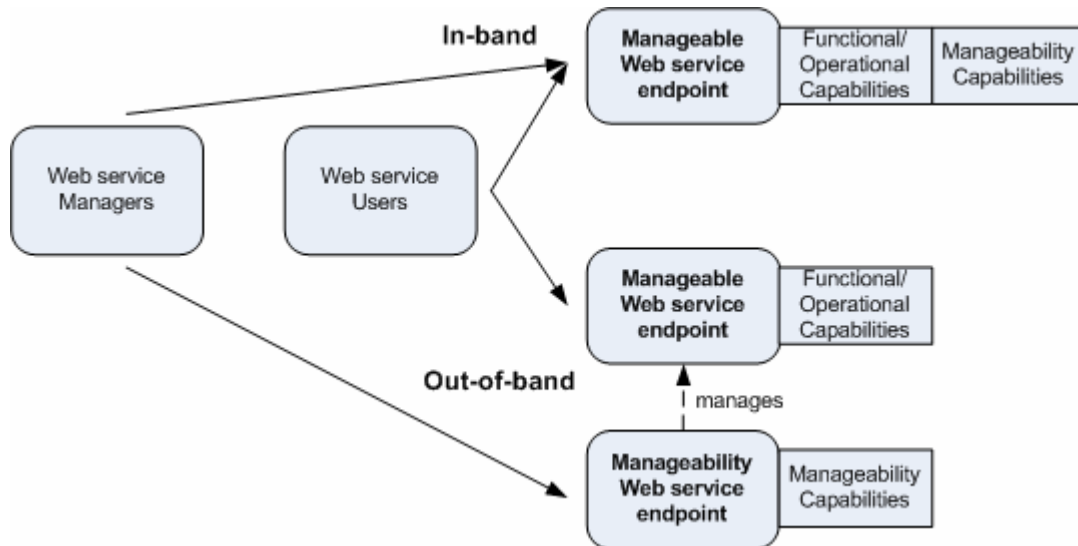
Refer to section 2 of the MUWS specification part 1 [MUWS] for more detailed explanation of discovery and message exchange between manageability consumers and manageability endpoints.

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

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

## 2.1 In-band and Out-of-band Manageability

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



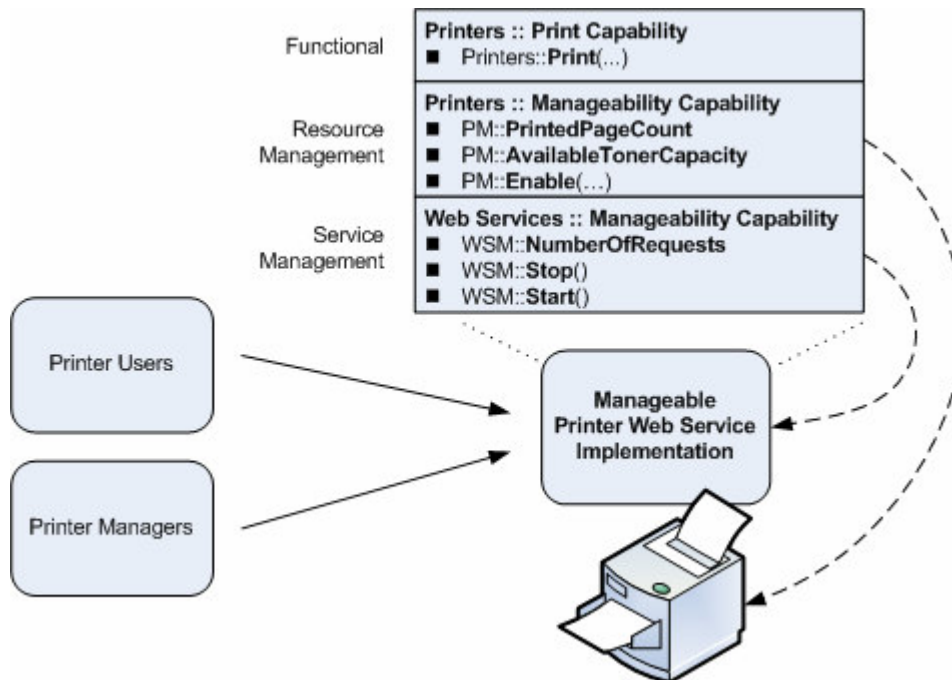
**Figure 2.** In-band and out-of-band manageability

## 2.2 Application to Resources Exposed as Web Services

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

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

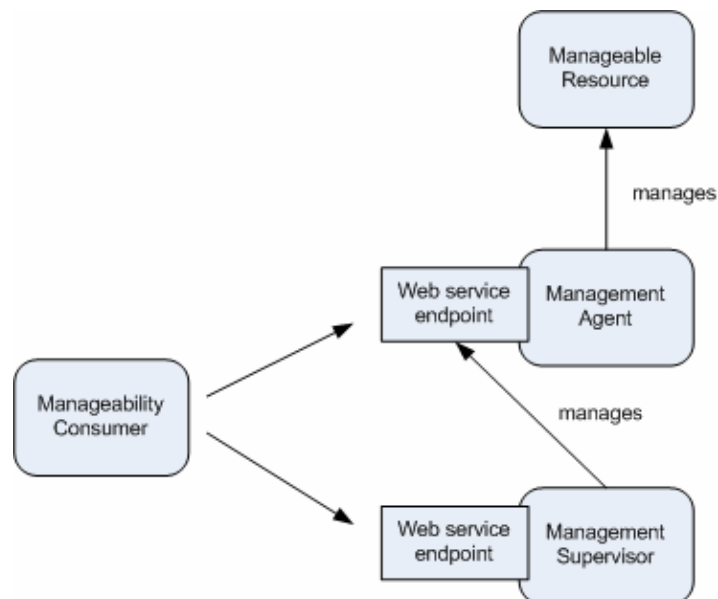
Composeability makes it easy for implementers of resource services to offer an appropriate set of functional capabilities along with an appropriate set of manageability capabilities guided by the appropriate model for authorization of these requests.



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

## 2.3 Self-Management

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



**Figure 4.**Applying MOWS to MUWS



---

## 3 Managing Web Services

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

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

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

### 3.1 Responsibilities of the Implementations of the Manageability Endpoints

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

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

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

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

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

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

## 3.2 Manageability at the Web service level

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

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

## 3.3 Using manageability of Web services endpoints

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

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

- 335  
336  
337  
338  
339  
340  
341
- b. Request the value of the ManageabilityCapability property (see MUWS specification part 1 section 5.2) and look for the URLs which identify the capabilities to be used.
  - 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.

---

## 4 Security Considerations

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

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

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

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

### 4.1 Additional security considerations when managing Web services

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

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

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

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

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

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

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

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

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

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

410

## 5 Web service manageability capabilities

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

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

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

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

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

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

### 5.1 Common manageability capabilities

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

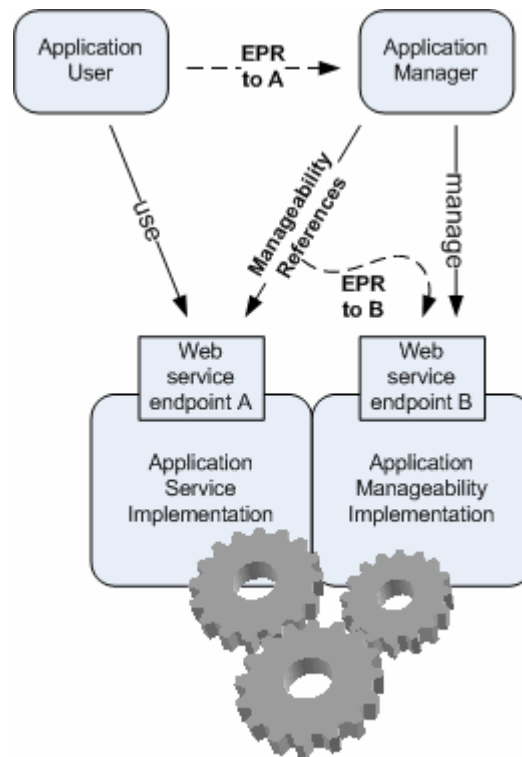
## 5.1.1 Manageability References

This capability is identified by the following URI:

<http://docs.oasis-open.org/mows2/capabilities/ManageabilityReferences>

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

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



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

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

- GetManageabilityReferences.

but applications may associate further operations with this capability.

### 5.1.1.1 Operations

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

#### 5.1.1.1.1 GetManageabilityReferences

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

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

```
<GetManageabilityReferences/>
```

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

The wsa:Action MUST contain the URI

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

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

```
<GetManageabilityReferencesResponse>
  <muws1:ManageabilityEndpointReference>
    <!-- see [MUWS] -->
  </muws1:ManageabilityEndpointReference>+
</GetManageabilityReferencesResponse>
```

The wsa:Action MUST contain the URI

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

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

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

## 5.2 Web service endpoint manageability capabilities

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

### 5.2.1 Identity

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



## 5.2.2 Identification

This capability is identified by the following URI:

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

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

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

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

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

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

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

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

This capability consists of two properties:

- EndpointReference (mandatory)
- EndpointDescriptions (optional)
  - Description (zero to many)

and defines one event: IdentificationCapability,

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

### 5.2.2.1 Properties

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

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

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

- Is not *Mutable*
- Is not *Modifiable*

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

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

### 545   5.2.2.2 Events

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

548

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

550

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

553

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

556

### 557   5.2.3 Metrics

558   This capability is identified by the following URI:

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

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

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

563

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

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

576   and defines one event: MetricsCapability,

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

578

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

582

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

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

### 5.2.3.1 Information markup declarations

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

```
<xs:complexType name="IntegerCounter">
  <xs:simpleContent>
    <xs:extension base="xs:nonNegativeInteger">
      <xs:attributeGroup ref="muws2:MetricAttributes"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

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

```
<xs:complexType name="DurationMetric">
  <xs:simpleContent>
    <xs:extension base="xs:duration">
      <xs:attributeGroup ref="muws2:MetricAttributes"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

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

### 5.2.3.2 Properties

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

```
<NumberOfRequests>IntegerCounter</NumberOfRequests>?
<NumberOfFailedRequests>IntegerCounter</NumberOfFailedRequests>?
<NumberOfSuccessfulRequests>IntegerCounter</NumberOfSuccessfulRequests>?
<ServiceTime>DurationMetric</ServiceTime>?
<MaxResponseTime>DurationMetric</MaxResponseTime>?
<LastResponseTime>DurationMetric</LastResponseTime>?
<MaxRequestSize>IntegerCounter</MaxRequestSize>?
<LastRequestSize>IntegerCounter</LastRequestSize>?
<MaxResponseSize>IntegerCounter</MaxResponseSize>?
<LastResponseSize>IntegerCounter</LastResponseSize>?
```

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

- Is *Mutable*
- Is not *Modifiable*
- <muws2:ChangeType>Counter</muws2:ChangeType>
- <muws2:TimeScope>SinceReset</muws2:TimeScope> or  
  <muws2:TimeScope>Interval</muws2:TimeScope>
- <muws2:GatheringTime>OnChange</muws2:GatheringTime>

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

- Is *Mutable*
- Is not *Modifiable*
- <muws2:ChangeType>Counter</muws2:ChangeType>
- <muws2:TimeScope>SinceReset</muws2:TimeScope> or  
  <muws2:TimeScope>Interval</muws2:TimeScope>
- <muws2:GatheringTime>OnChange</muws2:GatheringTime>

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

- Is *Mutable*
- Is not *Modifiable*
- <muws2:ChangeType>Counter</muws2:ChangeType>
- <muws2:TimeScope>SinceReset</muws2:TimeScope> or  
  <muws2:TimeScope>Interval</muws2:TimeScope>
- <muws2:GatheringTime>OnChange</muws2:GatheringTime>

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

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

- Is *Mutable*
- Is not *Modifiable*
- <muws2:ChangeType>Counter</muws2:ChangeType>
- <muws2:TimeScope>SinceReset</muws2:TimeScope> or  
  <muws2:TimeScope>Interval</muws2:TimeScope>
- <muws2:GatheringTime>OnChange</muws2:GatheringTime>

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

- Is *Mutable*
- Is not *Modifiable*

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

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

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

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

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

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

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

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

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

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

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

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

720

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

726

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

### 731   5.2.3.3 Events

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

734

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

736

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

739

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

742

### 743   5.2.4 Operation Metrics

744   This capability is identified by the following URI:

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

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

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

749

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

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

- MaxRequestSize (optional)
- LastRequestSize (optional)
- MaxResponseSize (optional)
- LastResponseSize (optional)

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

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

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

#### 5.2.4.1 Properties

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

```
<OperationMetrics operationName="xs:NCName", portType="xs:QName"? ...>*
  <NumberOfRequests>IntegerCounter</NumberOfRequests>?
  <NumberOfFailedRequests>IntegerCounter</NumberOfFailedRequests>?
  <NumberOfSuccessfulRequests>IntegerCounter</NumberOfSuccessfulRequests>?
  <ServiceTime>DurationMetric</ServiceTime>?
  <MaxResponseTime>DurationMetric</MaxResponseTime>?
  <LastResponseTime>DurationMetric</LastResponseTime> ?
  <MaxRequestSize>IntegerCounter</MaxRequestSize>?
  <LastRequestSize>IntegerCounter</LastRequestSize>?
  <MaxResponseSize>IntegerCounter</MaxResponseSize>?
  <LastResponseSize>IntegerCounter</LastResponseSize>?
  {any} *
</OperationMetrics>
```

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

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

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

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

#### 5.2.4.2 Events

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

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

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

#### 5.2.5 Operational State

This capability is identified by the following URI:

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

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

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

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

- **CurrentOperationalState** (mandatory)
- **LastOperationStateTransition** (zero to many)

and defines one event: **OperationalStateCapability**,

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

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

##### 5.2.5.1 Information markup declarations

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

- **UpState**

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

- **DownState**

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



847       ▪ **BusyState**

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

851       ▪ **IdleState**

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

855       ▪ **StoppedState**

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

860       ▪ **CrashedState**

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

864       ▪ **SaturatedState**

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

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

873  
874   The **OperationalStateType** XML Schema type is declared as follows.

```
875  
876   <xs:complexType name="OperationalStateType">  
877    <xs:complexContent>  
878      <xs:extension base="muws2:StateType"/>  
879    </xs:complexContent>  
880   </xs:complexType>
```

881  
882   The **OperationalStateType** is used to declare elements which contain any valid elements  
883   designating an operational state of a Web service endpoint.

- 884
- 885       ▪ A substate of the operational state MUST be declared according to the following rules.
    - 886           ○ An XML element is declared with a QName which identifies the desired substate  
887            semantics, for example my-app:DatabaseCleanupState
    - 888           ○ The contents of the XML element MUST be the only element which corresponds  
889            to the generalized state, for example mows:StoppedState

890  
891   An instance of the request processing state information represented in XML may look as shown  
892   in the following example,

```

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

```

### 5.2.5.2 Properties

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

```

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

```

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

- Is *Mutable*
- Is not *Modifiable*

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

- Is *Mutable*
- Is not *Modifiable*

### 5.2.5.3 Events

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

```

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

```

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

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

## 5.2.6 Operational Status

This capability is identified by the following URI:

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

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

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

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

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

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

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

- The WSLC **UP** state **MUST** be reported as the **Available** contents of the **muws2:OperationalStatus** property. Any sub-state of WSLC **UP** **MUST** be reported as **Available**.
- The WSLC **DOWN** state **MUST** be reported as the **Unavailable** contents of the **muws2:OperationalStatus** property. Any sub-state of WSLC **DOWN** **SHOULD** be reported as **Unavailable**. The **STOPPED** and **CRASHED** substates of WSLC **DOWN** **MUST** be reported as **Unavailable**.
- The WSLC **SATURATED** sub-state of **DOWN** **MAY** be reported as the **PartiallyAvailable** contents of the **muws2:OperationalStatus** property.

### 5.2.6.1 Events

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

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

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

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

## 5.2.7 Operation Operational Status

This capability is identified by the following URI:

**http://docs.oasis-open.org/wsdm/mows-2/capabilities/OperationOperationalStatus**

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

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

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

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

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

and the capability defines one event: *OperationOperationalStatusCapability*

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

### 5.2.7.1 Properties

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

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

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

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

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

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

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

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

1032

### 1033 5.2.7.2 Events

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

1036

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

1039

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

1042

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

1045

### 1046 5.2.8 Request Processing State

1047 This capability is identified by the following URI:

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

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

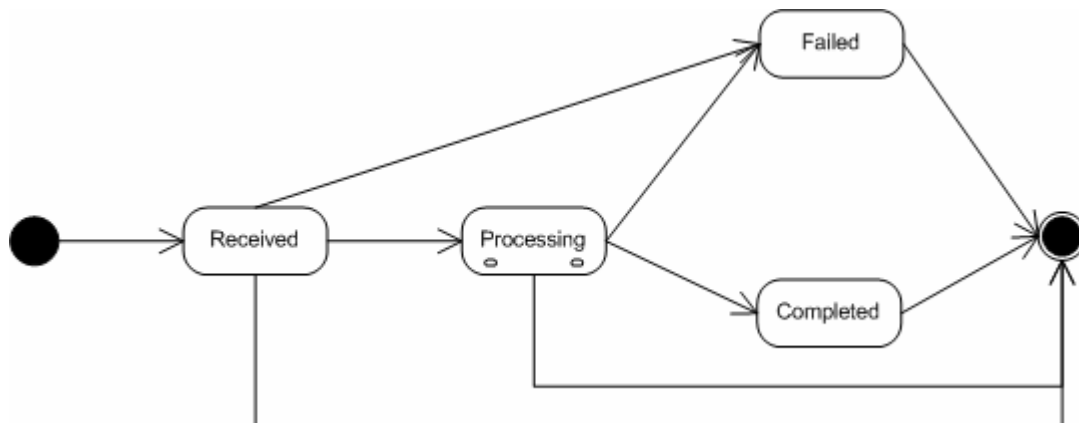
1050 • <muws2:Capability>[http://docs.oasis-open.org/mows-](http://docs.oasis-open.org/mows-2/capabilities/RequestProcessingState)  
1051 [2/capabilities/RequestProcessingState](http://docs.oasis-open.org/mows-2/capabilities/RequestProcessingState)</muws2:Capability>

1052

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

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

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



**Figure 6.** Request processing states

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

### 5.2.8.1 Information markup declarations

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

- **RequestReceivedState**

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.

- **RequestProcessingState**

This element corresponds to the Processing top-level state which means that the Web service endpoint is doing some internal processing/execution to fulfill the requested function. This state represents the earliest point at which the application module or business logic begins processing the request. For example, if the application server queues the request before dispatching it to the business logic, the time difference between "request received" and "processing" will include the duration the request was queued.

- **RequestCompletedState**

This element corresponds to the Completed top-level state which means that the Web service endpoint successfully completed requested function returning results to the requester.

1102       ▪ **RequestFailedState**

1103           This element corresponds to the Failed top-level state which means that the Web service  
1104           endpoint encountered an error and didn't complete the requested function, returning  
1105           error/fault to the requester.

1106  
1107       It is possible to extend the above state model. Substates of the Processing top-level state MAY  
1108       be introduced and MUST be identified by QNames, however, new top-level request processing  
1109       states MUST NOT be defined. In order to represent the taxonomy lineage of substates in XML,  
1110       the MUWS approach is used (section 3.2 in the [MUWS] part 2).

1111  
1112       The **RequestProcessingStateType** XML Schema type is declared as follows.

```
1113  
1114       <xs:complexType name="RequestProcessingStateType">  
1115       <xs:complexContent>  
1116           <xs:extension base="muws2:StateType"/>  
1117       </xs:complexContent>  
1118       </xs:complexType>
```

1119  
1120       The **RequestProcessingStateType** is used to declare elements which designate a request  
1121       processing state – top-level or substates of the Processing.

1122  
1123       A substate of the Processing compound state MUST be declared according to the following rules.

1124       An XML element is declared with a QName which identifies the desired substate semantics, for  
1125       example my-soap:SerializationState

1126       The contents of the XML element MUST be the only element which corresponds to the  
1127       generalized state, for example muws2:RequestProcessingState

1128  
1129       An instance of the request processing state information represented in XML may appear as  
1130       shown in the following example,

```
1131  
1132       <my:RequestProcessingStateInformationElement  
1133       xsi:type="mows:RequestProcessingStateType">  
1134           <my-soap:SerializationState>  
1135            <mows:RequestProcessingState/>  
1136           </my-soap:SerializationState>  
1137       </my:RequestProcessingStateInformationElement>
```

1138  
1139       **5.2.8.2 Events**

1140       Notifications are emitted when requests enter one of the request processing states (Figure 6).

1141  
1142       Property change events MUST be wrapped in Management events and published on topics  
1143       defined in this section.

1144  
1145       The following specification defines the Web service endpoint request processing state notification  
1146       topics in the namespace mapped to the **mowse** prefix. The message patterns' expression and  
1147       dialect MUST match precisely what is declared below.

1148

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



```

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

```

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

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

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

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

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

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

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

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

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

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

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

1304

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

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

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

#### 5.2.8.2.1 RequestProcessingNotification message

The RequestProcessingNotification message format is defined as follows.

```
<RequestProcessingNotification CurrentTime="xs:dateTime" ...>
<Request ...>
  <TransportInformation ...> {any}* </TransportInformation> ?
  <Message ...>
    <Size Unit=("bit" | "byte" | "word" | "dword" | "qword")
      ...>xs:positiveInteger</Size> ?
    (
      <NotIncluded/> |
      <Text>xs:string</Text> |
      <Binary>xs:base64Binary</Binary> |
      <Xml>{any}* </Xml>
    )
    {any}*
  </Message>
  {any}*
</Request> ?
<Reply ...>
  <!-- ... see contents of the Request element above ... -->
</Reply> ?
<StateInformation>
<muws2:StateTransition> <!-- ...see [MUWS]... --> </muws2:StateTransition> +
</StateInformation>
{any}*
</RequestProcessingNotification>
```

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

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

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

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

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

```
<TcpIpInfo
  Direction=("from" | "to")
  Port="xs:positiveInteger"
  Protocol=("TCP" | "UDP") ...>
(
  <IPv4Address>
    xs:hexBinary[xs:length[@value="8" and @fixed="true"]]
  </IPv4Address> |
  <IPv6Address>
    xs:hexBinary[xs:length[@value="32" and @fixed="true"]]
  </IPv6Address>
)
{any}*
</TcpIpInfo>
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**RequestProcessingNotification/StateInformation/muws2:StateTransition/muws2:EnteredState** indicates which request processing state was entered.

**RequestProcessingNotification/StateInformation/muws2:StateTransition/muws2:PreviousState** indicates which request processing state was exited.

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

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

#### 5.2.8.2.2 Examples of events against the Web service endpoint request processing state

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

Request:

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

```

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

```

1455

1456 Reply:

```

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

```

1466

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

1472

1473 Selector:

```

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

```

1476

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

1480

```

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

```

```

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

```

---

## 6 References

### 6.1 Normative

- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- [MUWS] V. Bullard, et al., *Web Services Distributed Management:Management using Web Services (MUWS 1.1) Part 1*, <http://docs.oasis-open.org/wsdm/wsdm-muws1-1.1-cd-01.pdf>
- V. Bullard, et al., *Web Services Distributed Management:Management using Web Services (MUWS 1.1) Part 2*, <http://docs.oasis-open.org/wsdm/wsdm-muws2-1.1-cd-01.pdf>
- [WS-A] Don Box, et al., *Web services Addressing (WS-Addressing)*, <http://www.w3.org/TR/ws-addr-core>
- [WS-RP] Steve Graham, et al., *Web Services Resource Properties 1.2 (WS-ResourceProperties)*, [http://docs.oasis-open.org/wsrf/wsrf-ws\\_resource\\_properties-1.2-spec-os-01.pdf](http://docs.oasis-open.org/wsrf/wsrf-ws_resource_properties-1.2-spec-os-01.pdf)
- [WS-N] Steve Graham, et al., *Web Services Base Notification 1.2 (WS-BaseNotification)*, [http://docs.oasis-open.org/wsn/wsn-ws\\_base\\_notification-1.3-spec-os.pdf](http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf)
- [WS-T] William Vambenepe, *Web Services Topics 1.2 (WS-Topics)*, [http://docs.oasis-open.org/wsn/wsn-ws\\_topics-1.3-spec-os.pdf](http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf)
- [WSDL] Erik Christensen, et al., *Web services Description Language (WSDL) 1.1*, W3C Note, March 2001, <http://www.w3.org/TR/wSDL>
- [SOAP] Don Box, et al., *Simple Object Access Protocol 1.1*, W3C Note, May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [XMLS] Henry S. Thompson, et al. *XML Schema Part 1: Structures*, W3C Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-1/>
- Paul V. Biron, et al. *XML Schema Part 2: Datatypes*, W3C Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-2/>
- [XML] Tim Bray, et al., *Extensible Markup Language (XML) 1.0 (Third Edition)*, W3C Recommendation, February 2004, <http://www.w3.org/TR/REC-xml>
- [XNS] Tim Bray, et al., *Namespaces in XML*, W3C Recommendation, January 1999, <http://www.w3.org/TR/REC-xml-names/>
- [XPath] James Clark, et al., *XML Path Language (XPath) Version 1.0*, W3C Recommendation, November 1999, <http://www.w3.org/TR/1999/REC-xpath-19991116>

### 6.2 Non-normative

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



1583	<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>
1584		
1585		
1586	<b>[BSP]</b>	Abbie Barbir, et al., <i>Basic Security Profile Version 1.0</i> , WS-I Working
1587		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-</a>
1588		<a href="http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0-2004-05-12.html">1.0-2004-05-12.html</a>
1589		

---

## Appendix A. Acknowledgments

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

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

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

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

## Appendix B. Revision History

Rev	Date	By Whom	
wd	2003-10-31	Igor Sedukhin	
wd	2003-11-14	Igor Sedukhin	
wd	2003-12-02	Igor Sedukhin	
wd	2004-01-26	Igor Sedukhin	
wd	2004-02-17	Igor Sedukhin	
wd	2004-03-01	Igor Sedukhin	
wd	2004-03-18	Igor Sedukhin	
wd	2004-03-19	Igor Sedukhin	
wd	2004-03-24	Igor Sedukhin	
wd	2004-03-24	Igor Sedukhin	
cd	2004-04-02	Igor Sedukhin	
wd	2004-07-21	Igor Sedukhin	
wd	2004-09-11	Igor Sedukhin	
wd	2004-10-11	Igor Sedukhin	
wd	2004-10-24	Igor Sedukhin	
wd	2004-11-04	Igor Sedukhin	
wd	2004-11-15	Igor Sedukhin	
wd	2004-11-19	Igor Sedukhin	
wd	2004-11-23	Igor Sedukhin	
wd	2004-12-03	Igor Sedukhin	
cd	2004-12-10	Igor Sedukhin	
standard	2005-03-09	Igor Sedukhin	
wd v 1.1	2005-10-27	Kirk Wilson	Addition of new capability: OperationMetric Creation of Appendix A
wd v 1.1 # 3	2005-11-03	Kirk Wilson	Addition of new Metric properties Version 1.1 namespace and file names added
wd v 1.1 #4	2005-11-04	Kirk Wilson	Addition of new capability: OperationOperationalStatus
wd v 1.1 #5	2005-11-16	Kirk Wilson	Restructure treatment of OperationMetrics and

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

1612

---

## Appendix C. Notices

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

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

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

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

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

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

## Appendix D. XML Schemas

```
1645 <?xml version="1.0" encoding="utf-8"?>
1646 <xs:schema targetNamespace="http://docs.oasis-open.org/wsdm/mows-2.xsd"
1647   xmlns:mows="http://docs.oasis-open.org/wsdm/mows-2.xsd"
1648   xmlns:muws2="http://docs.oasis-open.org/wsdm/muws2-2.xsd"
1649   xmlns:muws1="http://docs.oasis-open.org/wsdm/muws1-2.xsd"
1650   xmlns:wsa="http://www.w3.org/2005/08/addressing"
1651   xmlns:xs="http://www.w3.org/2001/XMLSchema"
1652   elementFormDefault="qualified" attributeFormDefault="unqualified">
1653
1654   <xs:import namespace="http://docs.oasis-open.org/wsdm/muws1-2.xsd"
1655 schemaLocation="http://docs.oasis-open.org/wsdm/muws1-2.xsd"/>
1656   <xs:import namespace="http://docs.oasis-open.org/wsdm/muws2-2.xsd"
1657 schemaLocation="http://docs.oasis-open.org/wsdm/muws2-2.xsd"/>
1658   <xs:import namespace="http://www.w3.org/2005/08/addressing"
1659 schemaLocation="http://www.w3.org/2005/08/addressing/ws-addr.xsd"/>
1660
1661   <!-- MOWS::ManageabilityReferences -->
1662   <xs:element name="GetManageabilityReferences"/>
1663   <xs:element name="GetManageabilityReferencesResponse">
1664     <xs:complexType>
1665       <xs:sequence>
1666         <xs:element ref="muws1:ManageabilityEndpointReference"
1667           maxOccurs="unbounded"/>
1668       </xs:sequence>
1669     </xs:complexType>
1670   </xs:element>
1671
1672   <!-- MOWS::Identification -->
1673   <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
1674   <xs:element name="EndpointDescriptions">
1675     <xs:complexType>
1676       <xs:sequence>
1677         <xs:element name="description" type="xs:anyURI"
1678           minOccurs="0" maxOccurs="unbounded"/>
1679       </xs:sequence>
1680       <xs:anyAttribute namespace="##other" processContents="lax"/>
1681     </xs:complexType>
1682   </xs:element>
1683
1684   <!-- Operation Properties Attribute Group -->
1685   <xs:attributeGroup name="OperationNameGroup">
1686     <xs:attribute name="operationName" type="xs:NCName" use="required"/>
1687     <xs:attribute name="portType" type="xs:QName" use="optional"/>
1688   </xs:attributeGroup>
1689
1690   <!-- MOWS Operation Operation Status Type -->
1691   <xs:complexType name="OperationOperationalStatusType">
1692     <xs:sequence>
1693       <xs:element ref="muws2:OperationalStatus" />
1694       <xs:any namespace="##any" processContents="lax"
1695         minOccurs="0" maxOccurs="unbounded"/>
1696     </xs:sequence>
1697   </xs:complexType>
```

```

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

```

```

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

```



```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

---

## Appendix E. WSDL elements

```
<?xml version="1.0" encoding="utf-8"?>
<w:definitions xmlns:w="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:wsrf-rp="http://docs.oasis-open.org/wsrf/rp-2"
  xmlns:mows="http://docs.oasis-open.org/wsdm/mows-2.xsd"
  xmlns:mowsw="http://docs.oasis-open.org/wsdm/mows-2.wsdl"
  targetNamespace="http://docs.oasis-open.org/wsdm/mows-2.wsdl">
  <w:types>
    <xs:import namespace="http://docs.oasis-open.org/wsdm/mows-2.xsd"
      schemaLocation="http://docs.oasis-open.org/wsdm/mows-2.xsd"/>
  </w:types>

  <w:message name="GetManageabilityReferencesRequest">
    <w:part name="body" element="mows:GetManageabilityReferences"/>
  </w:message>
  <w:message name="GetManageabilityReferencesResponse">
    <w:part name="body"
      element="mows:GetManageabilityReferencesResponse"/>
  </w:message>

</w:definitions>

<!--
      WSDL COPY Material
  Cut and Paste the operation specification below into a portType definition of the WSDL
  documents of a web service.

  NOTE: You must import the MOWS WSDL (mowsw).

  <operation name="GetManageabilityReferences">
    <input name="GetManageabilityReferencesRequest"
      message="mowsw:GetManageabilityReferencesRequest"/>
    <output name="GetManageabilityReferencesResponse"
      message="mowsw:GetManageabilityReferencesResponse"/>
  </operation>

-->
```

## Appendix F. Notification topic spaces

```
<?xml version="1.0" encoding="utf-8"?>
<wstop:TopicSpace name="MOWS"
  targetNamespace="http://docs.oasis-open.org/wsdm/mowse-2.xml"
  xmlns:muws2="http://docs.oasis-open.org/wsdm/muws2-2.xsd"
  xmlns:muws1="http://docs.oasis-open.org/wsdm/muws1-2.xsd"
  xmlns:mows="http://docs.oasis-open.org/wsdm/mows-2.xsd"
  xmlns:wstop="http://docs.oasis-open.org/wsn/t-1">

  <wstop:Topic name="IdentificationCapability" messageTypes="muws1:ManagementEvent"/>
  <wstop:Topic name="MetricsCapability" messageTypes="muws1:ManagementEvent"/>
  <wstop:Topic name="OperationMetricsCapability" messageTypes="muws1:ManagementEvent"/>
  <wstop:Topic name="OperationalStateCapability" messageTypes="muws1:ManagementEvent"/>
  <wstop:Topic name="OperationalStatusCapability"
    messageTypes="muws1:ManagementEvent"/>
  <wstop:Topic name="OperationOperationalStatusCapability"
    messageTypes="muws1:ManagementEvent"/>
  <wstop:Topic name="RequestProcessingStateCapability"
    messageTypes="muws1:ManagementEvent"/>

  <wstop:Topic name="RequestProcessingObservations"
    messageTypes="muws1:ManagementEvent">
    <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
      //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
        and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
    </wstop:MessagePattern>
    <wstop:Topic name="RequestReceived"
      messageTypes="muws1:ManagementEvent">
      <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
        //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
          and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
      </wstop:MessagePattern>
    </wstop:Topic>
    <wstop:Topic name="RequestProcessing"
      messageTypes="muws1:ManagementEvent">
      <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
        //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
          and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
      </wstop:MessagePattern>
    </wstop:Topic>
    <wstop:Topic name="RequestCompleted"
      messageTypes="muws1:ManagementEvent">
      <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
        //muws1:ManagementEvent[muws2:Situation/muws2:SituationCategory//muws2:ReportSituation
          and muws2:Severity="1" and count(mows:RequestProcessingNotification)=1]
      </wstop:MessagePattern>
    </wstop:Topic>
    <wstop:Topic name="RequestFailed"
      messageTypes="muws1:ManagementEvent">
      <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
```

```

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

```



2190 </wstop:TopicSpace>