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

Committee Draft, 10 December 2004

Document identifier:

cd-wsdm-mows-1.0

Location:

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

Editors:

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

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

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

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

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

Table of Contents

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 which 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 (§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.

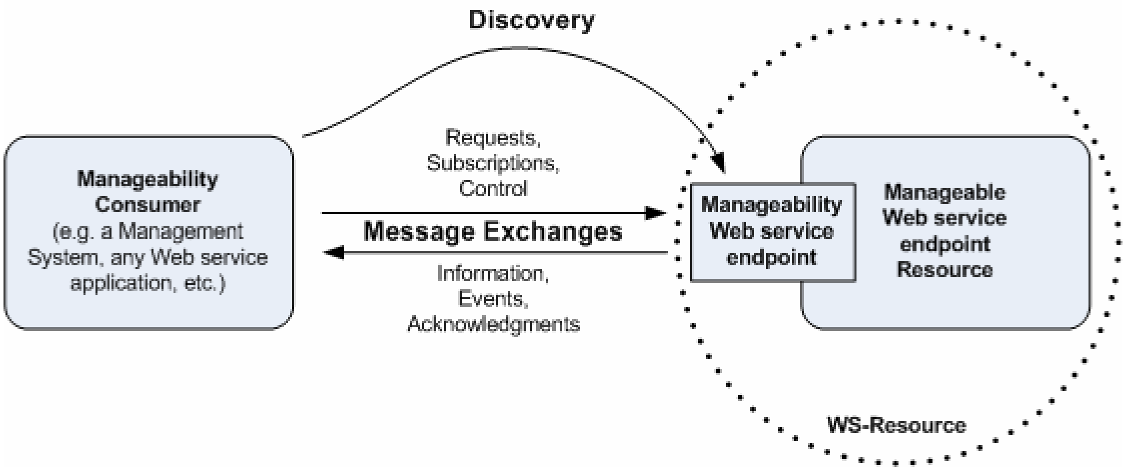

Discovery
Manageability Consumer
(e.g. a Management System, any Web service application, etc.)
Requests, Subscriptions, Control
Message Exchanges
Information, Events, Acknowledgments
Manageability Web service endpoint
Manageable Web service endpoint Resource
WS-Resource

Figure 1. Management of Web services concepts

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

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

- ***Focus on resources*** (§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*** (§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

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

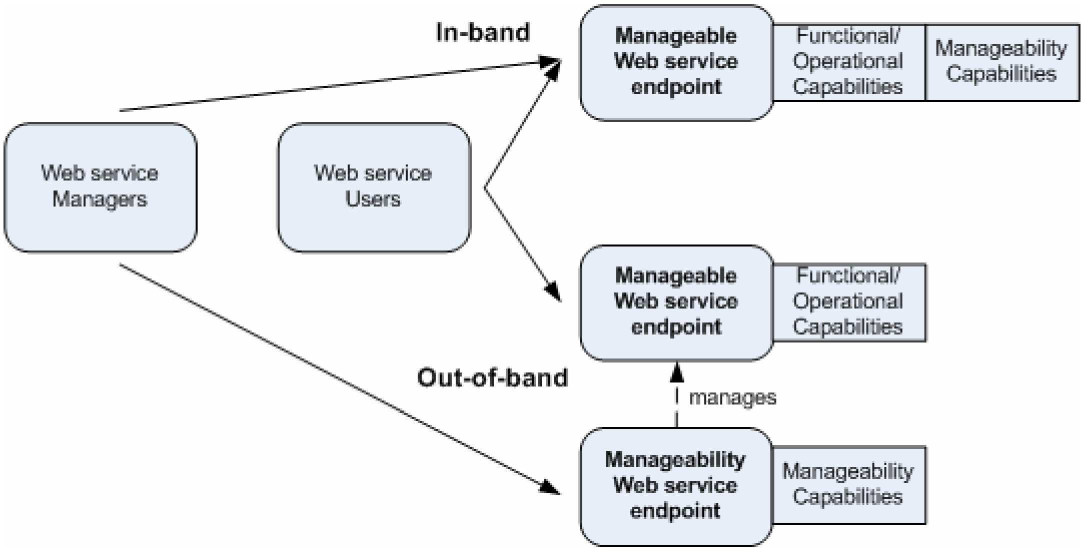

In-band
Manageable Web service endpoint
Functional/ Operational Capabilities
Manageability Capabilities
Web service Managers
Web service Users
Manageable Web service endpoint
Functional/ Operational Capabilities
Out-of-band
manages
Manageability Web service endpoint
Manageability Capabilities

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 good 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.

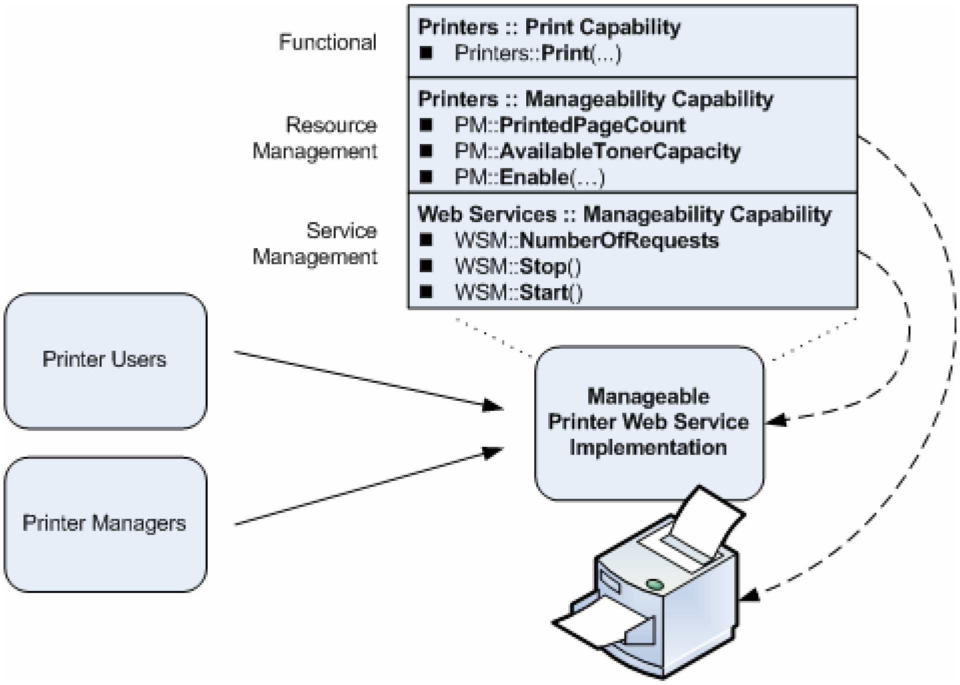


Figure 3. Application to resources exposed as Web services

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).

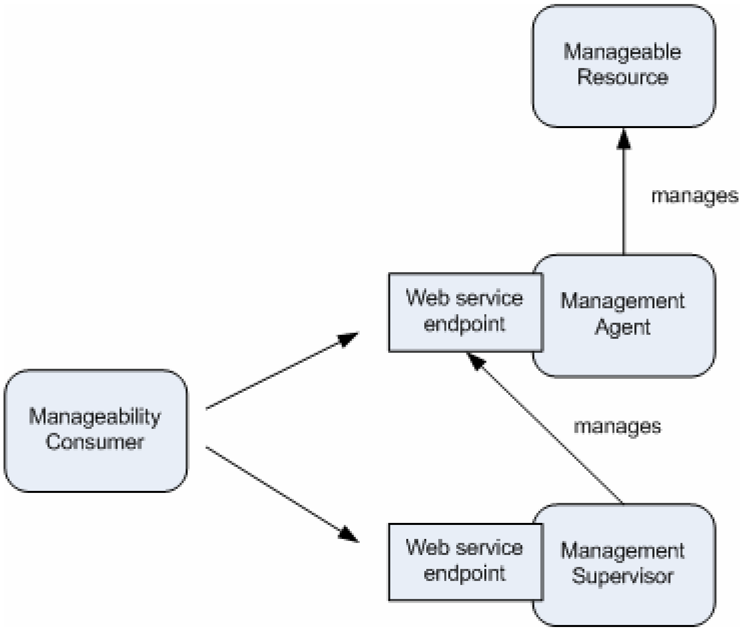


Figure 4. Applying MOWS to MUWS

2.4 Formal Representation of the Architecture

The following UML 2.0 model captures WSDM MOWS concepts in the context of the WSDM MUWS specifications **[MUWS]**. The diagram below is essentially a conceptual "mind map" or a digest of everything that has been described in the Architecture section.

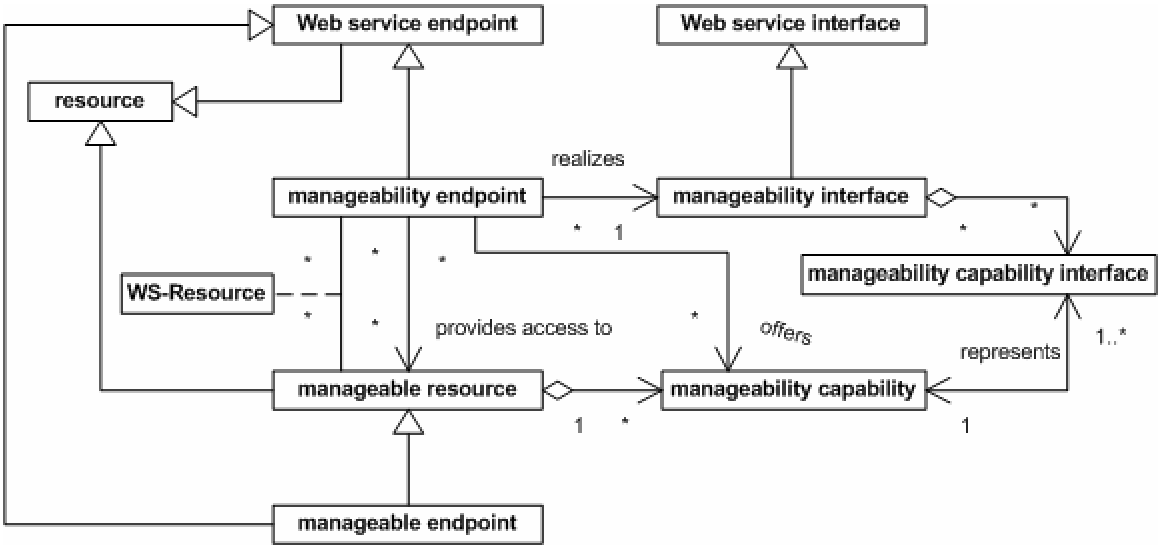


Figure 5. Formal expression of the Management of Web services architecture concepts

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 §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 §3.1. Aspects of (2) are further discussed in §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 wherein 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 (§5.1.1) on the functional Web services endpoint.
    c. The functional Web services endpoint may also be the manageability endpoint (§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 §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 §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.
    b. Request the value of the ManageabilityCapability property (see MUWS specification part 1 §5.2) and look for the URIs 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 WS-Trust. 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 §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 (§2.3), it will be important to follow a set of guidelines and best practices that detail how

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

- Implementers of this specification may need to give a particular attention to security when implementing the following manageability capabilities.
  - Manageability References (§5.1.1) – this capability allows access to the manageability endpoint references of a functional Web service endpoint. The concern is that visibility to these references may need to be protected differently than visibility of the functional Web service endpoint and its operations.
  - Request Processing State (§5.2.6) – this capability allows managers to subscribe to notifications against request processing by a functional Web service endpoint.
    1. Not all managers should be allowed to subscribe to request processing notification because messages may contain protected information, and/or may be used to generate a DoS attack.
    2. The request messages may be encrypted and signed. Therefore, managers may need to possess information that allows them to deal with such encrypted and signed messages.
    3. Notification messages which contain information about request messages SHOULD be encrypted to avoid spoofing of this information by intercepting notification messages.
    4. The request processing notification message provides sufficient flexibility with respect to its content to avoid inclusion of information which needs to be highly protected and therefore not relayed to managers.

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 §3.4 and part 2 §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. 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
muws-xs1	http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part1.xsd
muws-xs2	http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part2.xsd
muws-wsdl	http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part2.wsdl
mows-xs	http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.xsd
mows-wsdl	http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.wsdl
mows-events	http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows-events.xml
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing
wsdl	http://www.w3.org/2002/07/wsdl
S	http://www.w3.org/2002/12/soap-envelope
xs	http://www.w3.org/2001/XMLSchema
wsrf-rp	http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.xsd
wsnt	http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.xsd
wstop	http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.xsd

Unless otherwise specified, XML elements and XML schema types introduced in this specification belong to the namespace mapped to the **mows-xs** 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/wsdm/2004/12/mows/capabilities/ManageabilityReferences

This capability allows a functional/operational Web service or a resource exposed as a Web service (§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 6), 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.

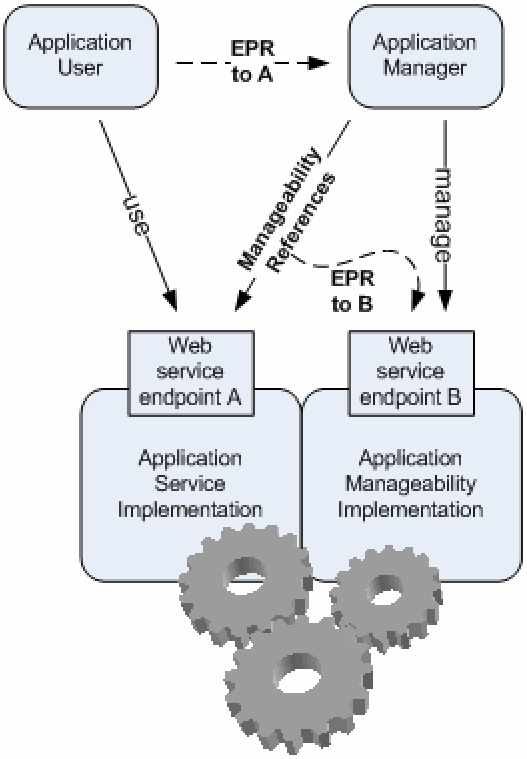


Figure 6. Use of Manageability References capability

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

MOWS:ManageabilityReferences
GetManageabilityReferences()

Figure 7. Manageability References capability model

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

```
<GetManageabilityReferencesResponse>
<muws-xs1:ManageabilityEndpointReference>
        <!-- see [MUWS] -->
</muws-xs1:ManageabilityEndpointReference>+
</GetManageabilityReferencesResponse>
```

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

GetManageabilityReferencesResponse/muws-xs1: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 (§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/wsdm/2004/12/mows/capabilities/Identification

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

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

The Web service endpoint's manageable identification capability is represented by the **Identification** UML model class. The name of the class identifies the semantics of this capability. 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 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.

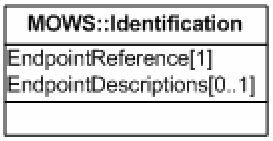


Figure 8. Endpoint identification manageability capability model

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.

- Is *Mutable*
- Is not *Modifiable*

5.2.2.2 Events

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

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

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

5.2.3 Metrics

This capability is identified by the following URI:
http://docs.oasis-open.org/wsdm/2004/12/mows/capabilities/Metrics

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

- <muws-xs2:Capability>http://docs.oasis-open.org/wsdm/2004/12/mows/capabilities/Metrics</muws-xs2:Capability>

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

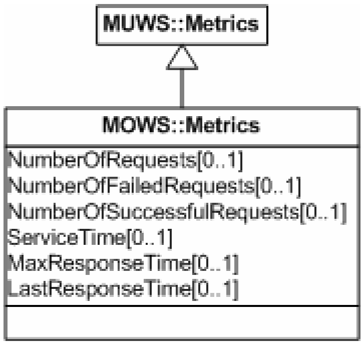


Figure 9. Endpoint metrics manageability capability model

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

It is recommended that for adequate calculations, the Web service endpoint metric properties (one or all) are retrieved together with the **muws-xs2: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 §5.2.6.

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="muws-xs2: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="muws-xs2: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>?
```

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 Figure 12. Metadata about this property is as follows.

- Is *Mutable*
- Is not *Modifiable*
- <muws-xs2:ChangeType>Counter</muws-xs2:ChangeType>
- <muws-xs2:TimeScope>SinceReset</muws-xs2:TimeScope> or <muws-xs2:TimeScope>Interval</muws-xs2:TimeScope>

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 Figure 12. Metadata about this property is as follows.

- Is *Mutable*
- Is not *Modifiable*
- <muws-xs2:ChangeType>Counter</muws-xs2:ChangeType>
- <muws-xs2:TimeScope>SinceReset</muws-xs2:TimeScope> or <muws-xs2:TimeScope>Interval</muws-xs2:TimeScope>

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 Figure 12. Metadata about this property is as follows.

- Is *Mutable*
- Is not *Modifiable*
- <muws-xs2:ChangeType>Counter</muws-xs2:ChangeType>
- <muws-xs2:TimeScope>SinceReset</muws-xs2:TimeScope> or <muws-xs2:TimeScope>Interval</muws-xs2:TimeScope>

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 (in seconds) 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*
- <muws-xs2:ChangeType>Counter</muws-xs2:ChangeType>
- <muws-xs2:TimeScope>SinceReset</muws-xs2:TimeScope> or <muws-xs2:TimeScope>Interval</muws-xs2:TimeScope>

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

- Is *Mutable*
- Is not *Modifiable*
- <muws-xs2:ChangeType>Gauge</muws-xs2:ChangeType>
- <muws-xs2:TimeScope>SinceReset</muws-xs2:TimeScope> or <muws-xs2:TimeScope>Interval</muws-xs2:TimeScope>

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

- Is *Mutable*
- Is not *Modifiable*
- <muws-xs2:ChangeType>Gauge</muws-xs2:ChangeType>
- <muws-xs2:TimeScope>PointInTime</muws-xs2:TimeScope>

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

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

5.2.3.3 Events

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

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

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

5.2.4 Operational State

This capability is identified by the following URI:
http://docs.oasis-open.org/wsdm/2004/12/mows/capabilities/OperationalState

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

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

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

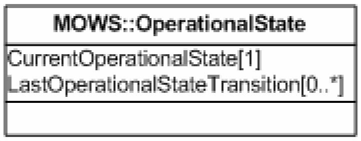

MOWS::OperationalState
CurrentOperationalState[1]
LastOperationalStateTransition[0..*]

Figure 10. Endpoint operational state manageability capability model

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.

5.2.4.1 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 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 is capable of accepting new requests. This element may contain the BUSY and IDLE substates of UP, as defined below.

- **DownState**

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

- **BusyState**

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

- **IdleState**

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

- **StoppedState**

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

- **CrashedState**

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

- **SaturatedState**

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

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

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

```
<xs:complexType name=”OperationalStateType”>
<xs:complexContent>
        <xs:extension base=”muws-xs2:StateType”/>
</xs:complexContent>
</xs:complexType>
```

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

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

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

```
<my:OperationalStateInformationElement xsi:type=”mows-xs:OperationalStateType”>
        <my-app:DatabaseCleanupState>
                <mows-xs:StoppedState>
                        <mows-xs:DownState/>
                </mows-xs:StoppedState>
        </my-app:DatabaseCleanupState>
</my:RequestProcessingStateInformationElement>
```

5.2.4.2 Properties

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

```
<CurrentOperationalState>mows-xs:OperationalStateType</CurrentOperationalState>
<LastOperationalStateTransition>
        muws-xs2:StateTransitionType
</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*

5.2.4.3 Events

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

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

mows-events: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.

5.2.5 Operational Status

This capability is identified by the following URI: **http://docs.oasis-open.org/wsdm/2004/12/mows/capabilities/OperationalStatus**

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

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

WSDM manageable endpoints that intend to support the MUWS **Operational Status** manageability capability (§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 **muws-xs2: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 **muws-xs2: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 **muws-xs2:OperationalStatus** property.

5.2.5.1 Events

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

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

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

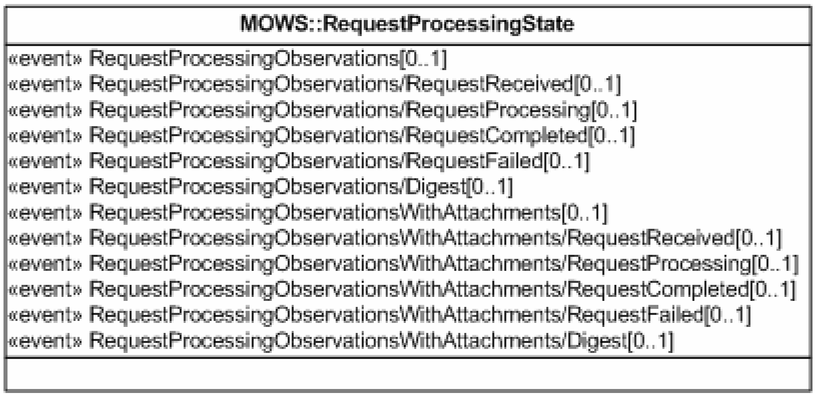
5.2.6 Request Processing State

This capability is identified by the following URI: **http://docs.oasis-open.org/wsdm/2004/12/mows/capabilities/RequestProcessingState**

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

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

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



MOWS::RequestProcessingState

«event» RequestProcessingObservations[0..1]
«event» RequestProcessingObservations/RequestReceived[0..1]
«event» RequestProcessingObservations/RequestProcessing[0..1]
«event» RequestProcessingObservations/RequestCompleted[0..1]
«event» RequestProcessingObservations/RequestFailed[0..1]
«event» RequestProcessingObservations/Digest[0..1]
«event» RequestProcessingObservationsWithAttachments[0..1]
«event» RequestProcessingObservationsWithAttachments/RequestReceived[0..1]
«event» RequestProcessingObservationsWithAttachments/RequestProcessing[0..1]
«event» RequestProcessingObservationsWithAttachments/RequestCompleted[0..1]
«event» RequestProcessingObservationsWithAttachments/RequestFailed[0..1]
«event» RequestProcessingObservationsWithAttachments/Digest[0..1]

Figure 11. Endpoint request processing state manageability capability model

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.

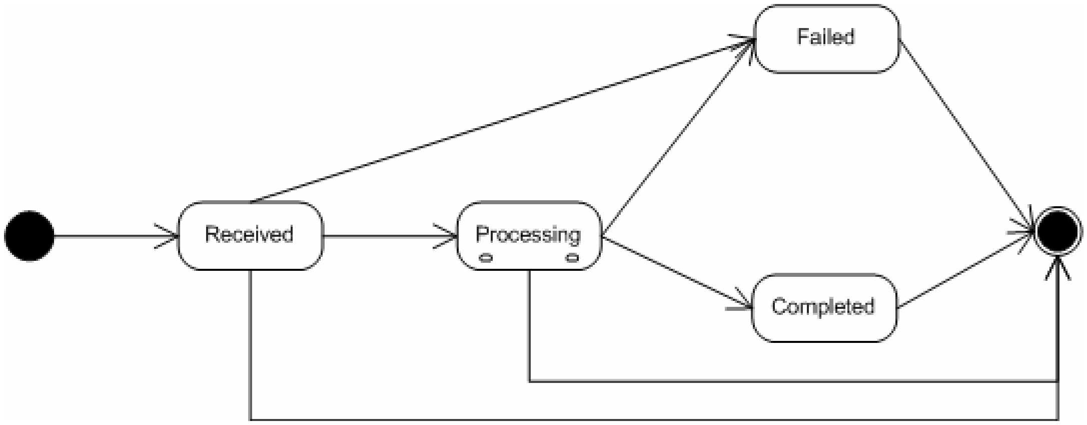


Figure 12. Request processing states

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.

5.2.6.1 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 12).

- **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.

- **RequestFailedState**

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

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

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

```
<xs:complexType name=”RequestProcessingStateType”>
<xs:complexContent>
        <xs:extension base=”muws-xs2:StateType”/>
</xs:complexContent>
</xs:complexType>
```

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

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

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

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

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

```
<my:RequestProcessingStateInformationElement xsi:type=”mows-
xs:RequestProcessingStateType”>
        <my-soap:SerializationState>
                <mows-xs:RequestProcessingState/>
        </my-soap:SerializationState>
</my:RequestProcessingStateInformationElement>
```

5.2.6.2 Events

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

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

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

<wstop:Topic name="RequestProcessingObservations"
        messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        <wstop:Topic name="RequestReceived"
                messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        </wstop:Topic>
        <wstop:Topic name="RequestProcessing"
                messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        </wstop:Topic>
        <wstop:Topic name="RequestCompleted"
                messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        </wstop:Topic>
        <wstop:Topic name="RequestFailed"
                messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        </wstop:Topic>
        <wstop:Topic name="Digest"
                messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
```

```
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        </wstop:Topic>
</wstop:Topic>

<wstop:Topic name="RequestProcessingObservationsWithAttachments"
        messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        <wstop:Topic name="RequestReceived"
                messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        </wstop:Topic>
        <wstop:Topic name="RequestProcessing"
                messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        </wstop:Topic>
        <wstop:Topic name="RequestCompleted"
                messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        </wstop:Topic>
        <wstop:Topic name="RequestFailed"
                messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        </wstop:Topic>
        <wstop:Topic name="Digest"
                messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        </wstop:Topic>
</wstop:Topic>
```

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

mows-events:ManageableEndpoint/mows-events:RequestProcessingObservations indicates availability of any information about the processing of any request by the Web service endpoint (Figure 12) 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 §5.2.6.2.1). The MUWS management event MUST also declare the event situation category with the muws-xs2: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.

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

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

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

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

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

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

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

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

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 §5.

5.2.6.2.1 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>
<muws-xs2:StateTransition> <!-- …see [MUWS]… --> </muws-xs2:StateTransition> +
</StateInformation>
{any}*
```

```
</RequestProcessingNotification>
```

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

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 serializeable 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/muws-xs2: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/muws-xs2:StateTransition/@muws-xs2:Time indicates time when the described transition occurred. Note that according to the request processing state model (Figure 12), 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/muws-xs2:StateTransition/muws-xs2:EnteredState indicates which request processing state was entered.

RequestProcessingNotification/StateInformation/muws-xs2:StateTransition/muws-xs2: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.

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

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

Request:

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

Reply:

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

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

Selector:

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

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

```
<RequestProcessingNotification CurrentTime=”…”>
<Request>
        <TransportInformation>
                <TcpIpInfo Direction=”from” Port=”2840” Protocol=”TCP”>
                        <IPV4Address>C0A80002</IPV4Address>
                </TcpIpInfo>
                <TcpIpInfo Direction=”to” Port=”80” Protocol=”TCP”>
                        <IPV4Address>C0A80003</IPV4Address>
                </TcpIpInfo>
        </TransportInformation>
        <Message>
        <Size Unit=”byte”>257</Size>
        <Xml>
        <S:Envelope xmlns:S=”…” xmlns:x=”…” …>
        . . .
        <S:Body>
                <x:Order>
                        <x:Item>123</x:Item>
                        <x:Quantity>10</x:Quantity>
                </x:Order>
        </S:Body>
        </S:Envelope>
        </Xml>
        </Message>
</Request>
<Reply>
        <TransportInformation>
                <TcpIpInfo Direction=”to” Port=”2840” Protocol=”TCP”>
                        <IPV4Address>C0A80002</IPV4Address>
                </TcpIpInfo>
                <TcpIpInfo Direction=”from” Port=”80” Protocol=”TCP”>
                        <IPV4Address>C0A80003</IPV4Address>
                </TcpIpInfo>
        </TransportInformation>
        <Message>
        <Size Unit=”byte”>232</Size>
        <Xml>
        <S:Envelope xmlns:S=”…” xmlns:x=”…” …>
        . . .
        <S:Body>
                <x:Shipped>
                        <x:Item>123</x:Item>
                        <x:Quantity>2</x:Quantity>
                </x:Shipped>
        </S:Body>
        </S:Envelope>
        </Xml>
        </Message>
</Reply>
<muws-xs2:StateTransition Time=”…”>
<muws-xs2:EnteredState/><RequestCompletedState/></muws-xs2:EnteredState>
<muws-xs2:PreviousState><RequestProcessingState/></muws-xs2:PreviousState>
</muws-xs2:StateTransition>
. . .
</RequestProcessingNotification>
```

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] William Vambenepe, *Web Services Distributed Management:Management using Web Services (MUWS 1.0) Part 1*, OASIS Committee Draft, December 2004, http://docs.oasis-open.org/wsdm/2004/12/cd-wsdm-muws-part1-1.0.pdf

William Vambenepe, *Web Services Distributed Management:Management using Web Services (MUWS 1.0) Part 2*, OASIS Committee Draft, December 2004, http://docs.oasis-open.org/wsdm/2004/12/cd-wsdm-muws-part2-1.0.pdf

[WS-A] Don Box, et al., *Web services Addressing (WS-Addressing)*, W3C Member Submission, August 2004, http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/

[WS-RP] Steve Graham, et al., *Web Services Resource Properties 1.2 (WS-ResourceProperties)*, OASIS Working Draft, June 2004, http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-04.pdf

[WS-N] Steve Graham, et al., *Web Services Base Notification 1.2 (WS-BaseNotification)*, OASIS Working Draft, June 2004, http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf

[WS-T] William Vambenepe, *Web Services Topics 1.2 (WS-Topics)*, OASIS Working Draft, Jully 2004, http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.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/

[WSS] Anthony Nadalin, et al., *Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)*, March 2004, OASIS Standard, http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

[BSP] Abbie Barbir, et al., *Basic Security Profile Version 1.0*, WS-I Working Group Draft, May 2004, http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0-2004-05-12.html

Appendix A. Acknowledgments

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

The following individuals were members of the committee when the specification 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.

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

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.

Appendix D. XML Schemas

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-
mows.xsd" xmlns:mows-xs="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.xsd"
xmlns:muws-xs2="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part2.xsd"
xmlns:muws-xs1="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part1.xsd"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">

    <xs:import namespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
part1.xsd" schemaLocation="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
part1.xsd"/>
    <xs:import namespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
part2.xsd" schemaLocation="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
part2.xsd"/>
    <xs:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>

    <!-- MOWS::ManageabilityReferences -->
    <xs:element name="GetManageabilityReferences"/>
    <xs:element name="GetManageabilityReferencesResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="muws-xs1:ManageabilityEndpointReference"
                    maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <!-- MOWS::Identification -->
    <xs:element name="EndpointReference" type="wsa:EndpointReferenceType"/>
    <xs:element name="EndpointDescriptions">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="description" type="xs:anyURI"
                    minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:anyAttribute namespace="##other" processContents="lax"/>
        </xs:complexType>
    </xs:element>

    <xs:complexType name="EndpointIdentificationPropertiesType">
        <xs:sequence>
            <xs:element ref="mows-xs:EndpointReference"/>
            <xs:element ref="mows-xs:EndpointDescriptions" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="EndpointIdentificationProperties"
        type="mows-xs:EndpointIdentificationPropertiesType"/>

    <!-- MOWS::Metrics -->
    <xs:complexType name="IntegerCounter">
        <xs:simpleContent>
            <xs:extension base="xs:nonNegativeInteger">
                <xs:attributeGroup ref="muws-xs2:MetricAttributes"/>
                <xs:anyAttribute namespace="##other" processContents="lax"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>

    <xs:complexType name="DurationMetric">
        <xs:simpleContent>
            <xs:extension base="xs:duration">
                <xs:attributeGroup ref="muws-xs2:MetricAttributes"/>
                <xs:anyAttribute namespace="##other" processContents="lax"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>

    <xs:element name="NumberOfRequests" type="mows-xs:IntegerCounter"/>
    <xs:element name="NumberOfSuccessfulRequests" type="mows-xs:IntegerCounter"/>
    <xs:element name="NumberOfFailedRequests" type="mows-xs:IntegerCounter"/>
    <xs:element name="ServiceTime" type="mows-xs:DurationMetric"/>
    <xs:element name="MaxResponseTime" type="mows-xs:DurationMetric"/>
    <xs:element name="LastResponseTime" type="mows-xs:DurationMetric"/>

    <xs:complexType name="EndpointMetricsPropertiesType">
        <xs:sequence>
            <xs:element ref="mows-xs:NumberOfRequests" minOccurs="0"/>
            <xs:element ref="mows-xs:NumberOfFailedRequests" minOccurs="0"/>
            <xs:element ref="mows-xs:NumberOfSuccessfulRequests"
                 minOccurs="0"/>
            <xs:element ref="mows-xs:ServiceTime" minOccurs="0"/>
            <xs:element ref="mows-xs:MaxResponseTime" minOccurs="0"/>
            <xs:element ref="mows-xs:LastResponseTime" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="EndpointMetricsProperties"
        type="mows-xs:EndpointMetricsPropertiesType"/>

    <!-- MOWS::OperationalState -->
    <xs:complexType name="OperationalStateType">
        <xs:complexContent>
            <xs:extension base="muws-xs2:StateType"/>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="UpState">
        <xs:complexType>
            <xs:complexContent>
                <xs:restriction base="mows-xs:OperationalStateType"/>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="IdleState">
        <xs:complexType>
            <xs:complexContent>
                <xs:restriction base="mows-xs:OperationalStateType">
```

```
                    <xs:sequence>
                        <xs:element ref="mows-xs:UpState"/>
                    </xs:sequence>
                </xs:restriction>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="BusyState">
        <xs:complexType>
            <xs:complexContent>
                <xs:restriction base="mows-xs:OperationalStateType">
                    <xs:sequence>
                        <xs:element ref="mows-xs:UpState"/>
                    </xs:sequence>
                </xs:restriction>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="DownState">
        <xs:complexType>
            <xs:complexContent>
                <xs:restriction base="mows-xs:OperationalStateType"/>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="StoppedState">
        <xs:complexType>
            <xs:complexContent>
                <xs:restriction base="mows-xs:OperationalStateType">
                    <xs:sequence>
                        <xs:element ref="mows-xs:DownState"/>
                    </xs:sequence>
                </xs:restriction>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="CrashedState">
        <xs:complexType>
            <xs:complexContent>
                <xs:restriction base="mows-xs:OperationalStateType">
                    <xs:sequence>
                        <xs:element ref="mows-xs:DownState"/>
                    </xs:sequence>
                </xs:restriction>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="SaturatedState">
        <xs:complexType>
            <xs:complexContent>
                <xs:restriction base="mows-xs:OperationalStateType">
                    <xs:sequence>
                        <xs:element ref="mows-xs:DownState"/>
                    </xs:sequence>
                </xs:restriction>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>

    <xs:element name="CurrentOperationalState" type="mows-xs:OperationalStateType"/>
    <xs:element name="LastOperationalStateTransition"
        type="muws-xs2:StateTransitionType"/>

    <xs:complexType name="EndpointOperationalStatePropertiesType">
        <xs:sequence>
            <xs:element ref="mows-xs:CurrentOperationalState"/>
            <xs:element ref="mows-xs:LastOperationalStateTransition"
                minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="EndpointOperationalStateProperties"
        type="mows-xs:EndpointOperationalStatePropertiesType"/>

    <!-- MOWS::RequestProcessingState -->
    <xs:complexType name="RequestProcessingStateType">
        <xs:complexContent>
            <xs:extension base="muws-xs2:StateType"/>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="RequestReceivedState">
        <xs:complexType>
            <xs:complexContent>
                <xs:restriction base="mows-xs:RequestProcessingStateType"/>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="RequestProcessingState">
        <xs:complexType>
            <xs:complexContent>
                <xs:restriction base="mows-xs:RequestProcessingStateType"/>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="RequestCompletedState">
        <xs:complexType>
            <xs:complexContent>
                <xs:restriction base="mows-xs:RequestProcessingStateType"/>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="RequestFailedState">
        <xs:complexType>
            <xs:complexContent>
                <xs:restriction base="mows-xs:RequestProcessingStateType"/>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="MessageContentNotIncludedFlag"/>
    <xs:simpleType name="MessageSizeUnitType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="bit"/>
            <xs:enumeration value="byte"/>
```

```
                        <xs:enumeration value="word"/>
                        <xs:enumeration value="dword"/>
                        <xs:enumeration value="qword"/>
                </xs:restriction>
        </xs:simpleType>
        <xs:complexType name="MessageContentSizeType">
                <xs:simpleContent>
                        <xs:extension base="xs:positiveInteger">
                                <xs:attribute name="Unit"
                                        type="mows-xs:MessageSizeUnitType" use="required"/>
                                <xs:anyAttribute namespace="##other" processContents="lax"/>
                        </xs:extension>
                </xs:simpleContent>
        </xs:complexType>
        <xs:complexType name="MessageContentType">
                <xs:sequence>
                        <xs:element name="Size"
                                type="mows-xs:MessageContentSizeType" minOccurs="0"/>
                        <xs:choice>
                                <xs:element name="NotIncluded"
                                        type="mows-xs:MessageContentNotIncludedFlag"/>
                                <xs:element name="Text" type="xs:string"/>
                                <xs:element name="Binary" type="xs:base64Binary"/>
                                <xs:element name="Xml"
                                        type="mows-xs:AnyXmlContentsType"/>
                        </xs:choice>
                        <xs:any namespace="##other" processContents="lax"
                                minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:anyAttribute namespace="##other" processContents="lax"/>
        </xs:complexType>
        <xs:complexType name="AnyXmlContentsType">
                <xs:sequence>
                        <xs:any namespace="##any" processContents="lax"
                                minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:anyAttribute namespace="##any" processContents="lax"/>
        </xs:complexType>
        <xs:complexType name="MessageInformationType">
                <xs:sequence>
                        <xs:element name="TransportInformation"
                                type="mows-xs:AnyXmlContentsType" minOccurs="0"/>
                        <xs:element name="Message" type="mows-xs:MessageContentType"/>
                        <xs:any namespace="##any" processContents="lax"
                                minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:anyAttribute namespace="##any" processContents="lax"/>
        </xs:complexType>
        <xs:complexType name="RequestProcessingStateInformationType">
          <xs:sequence>
            <xs:element ref="muws-xs2:StateTransition" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
        <xs:element name="RequestProcessingNotification">
                <xs:complexType>
                        <xs:sequence>
                <xs:element name="Request"
                    type="mows-xs:MessageInformationType"
                    minOccurs="0"/>
                <xs:element name="Reply"
                    type="mows-xs:MessageInformationType"
                    minOccurs="0"/>
                <xs:element name="StateInformation"
                    type="mows-
xs:RequestProcessingStateInformationType"/>
                <xs:any namespace="##any" processContents="lax"
                    minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="CurrentTime" type="xs:dateTime" use="required"/>
            <xs:anyAttribute namespace="##any" processContents="lax"/>
        </xs:complexType>
    </xs:element>
    <xs:simpleType name="IPV4AddressType">
        <xs:restriction base="xs:hexBinary">
            <xs:length value="8" fixed="true"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:element name="IPV4Address" type="mows-xs:IPV4AddressType"/>
    <xs:simpleType name="IPV6AddressType">
        <xs:restriction base="xs:hexBinary">
            <xs:length value="32" fixed="true"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:element name="IPV6Address" type="mows-xs:IPV6AddressType"/>
    <xs:simpleType name="TcpIpDirectionType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="to"/>
            <xs:enumeration value="from"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="TcpIpProtocolType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="TCP"/>
            <xs:enumeration value="UDP"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:element name="TcpIpInfo">
        <xs:complexType>
            <xs:sequence>
                <xs:choice>
                    <xs:element ref="mows-xs:IPV4Address"/>
                    <xs:element ref="mows-xs:IPV6Address"/>
                </xs:choice>
                <xs:any namespace="##any" processContents="lax"
                     minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="Direction" type="mows-xs:TcpIpDirectionType"
                    use="required"/>
            <xs:attribute name="Port" type="xs:positiveInteger" use="required"/>
            <xs:attribute name="Protocol" type="mows-xs:TcpIpProtocolType"
                    use="required"/>
            <xs:anyAttribute namespace="##any" processContents="lax"/>
```

```
                </xs:complexType>
        </xs:element>
</xs:schema>
```

Appendix E. WSDL elements

```
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:wsrf-rp="http://docs.oasis-
open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.xsd" xmlns:mows-
xs="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.xsd" xmlns:mows-
wsdl="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.wsdl"
targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.wsdl">

        <types>
                <xs:schema elementFormDefault="qualified"
targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.wsdl">
                        <xs:import namespace="http://docs.oasis-
open.org/wsdm/2004/12/mows/wsdm-mows.xsd" schemaLocation="http://docs.oasis-
open.org/wsdm/2004/12/mows/wsdm-mows.xsd"/>
                </xs:schema>
        </types>

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

        <portType name="ManageabilityReferences">
                <operation name="GetManageabilityReferences">
                <input name="GetManageabilityReferencesRequest"
                        message="mows-wsdl:GetManageabilityReferencesRequest"/>
                <output name="GetManageabilityReferencesResponse"
                        message="mows-wsdl:GetManageabilityReferencesResponse"/>
                </operation>
        </portType>

        <portType name="EndpointIdentification"
                wsrf-rp:ResourceProperties="mows-xs:EndpointIdentificationProperties"/>

        <portType name="EndpointMetrics"
                wsrf-rp:ResourceProperties="mows-xs:EndpointMetricsProperties"/>

        <portType name="EndpointOperationalState"
                wsrf-rp:ResourceProperties="mows-xs:EndpointOperationalStateProperties"/>
</definitions>
```

Appendix F. Notification topic spaces

```
<?xml version="1.0" encoding="utf-8"?>
<wstop:TopicSpace name="MOWS"
        targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows-
events.xml"
        xmlns:muws-xs2="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
part2.xsd"
        xmlns:muws-xs1="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
part1.xsd"
        xmlns:mows-xs="http://docs.oasis-open.org/wsdm/2004/12/mows/wsdm-mows.xsd"
        xmlns:wstop="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.xsd">

<wstop:Topic name="IdentificationCapability" messageTypes="muws-xs1:ManagementEvent"/>
<wstop:Topic name="MetricsCapability" messageTypes="muws-xs1:ManagementEvent"/>
<wstop:Topic name="OperationalStateCapability" messageTypes="muws-
xs1:ManagementEvent"/>
<wstop:Topic name="OperationalStatusCapability" messageTypes="muws-
xs1:ManagementEvent"/>
<wstop:Topic name="RequestProcessingStateCapability" messageTypes="muws-
xs1:ManagementEvent"/>


<wstop:Topic name="RequestProcessingObservations"
        messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        <wstop:Topic name="RequestReceived"
                messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        </wstop:Topic>
        <wstop:Topic name="RequestProcessing"
                messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        </wstop:Topic>
        <wstop:Topic name="RequestCompleted"
                messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
```

```
    </wstop:Topic>
    <wstop:Topic name="RequestFailed"
        messageTypes="muws-xs1:ManagementEvent">
    <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
    </wstop:MessagePattern>
    </wstop:Topic>
    <wstop:Topic name="Digest"
        messageTypes="muws-xs1:ManagementEvent">
    <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
    </wstop:MessagePattern>
    </wstop:Topic>
</wstop:Topic>

<wstop:Topic name="RequestProcessingObservationsWithAttachments"
    messageTypes="muws-xs1:ManagementEvent">
    <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
    </wstop:MessagePattern>
    <wstop:Topic name="RequestReceived"
        messageTypes="muws-xs1:ManagementEvent">
    <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
    </wstop:MessagePattern>
    </wstop:Topic>
    <wstop:Topic name="RequestProcessing"
        messageTypes="muws-xs1:ManagementEvent">
    <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
    </wstop:MessagePattern>
    </wstop:Topic>
    <wstop:Topic name="RequestCompleted"
        messageTypes="muws-xs1:ManagementEvent">
    <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
    </wstop:MessagePattern>
    </wstop:Topic>
    <wstop:Topic name="RequestFailed"
        messageTypes="muws-xs1:ManagementEvent">
    <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
```

```
        </wstop:MessagePattern>
        </wstop:Topic>
        <wstop:Topic name="Digest"
                messageTypes="muws-xs1:ManagementEvent">
        <wstop:MessagePattern Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
//muws-xs1:ManagementEvent[muws-xs2:Situation/muws-xs2:SituationCategory//muws-
xs2:ReportSituation and muws-xs2:Severity="1" and count(mows-
xs:RequestProcessingNotification)=1]
        </wstop:MessagePattern>
        </wstop:Topic>
</wstop:Topic>
</wstop:TopicSpace>
```