Web Services Distributed Management: Management Using Web Services (MUWS 1.0) Part 1

Committee Draft, January 11th 2005

Document identifier:

cd-wsdm-muws-part1-1.0

Location:

http://docs.oasis-open.org/wsdm/2004/12/cd-wsdm-muws-part1-1.0.pdf

Editor:

William Vambenepe, Hewlett-Packard <vbp@hp.com>

Abstract:

There are two specifications produced by the Web Services Distributed Management technical committee: Management *Using* Web services (MUWS) and Management *Of* Web services (MOWS, see [MOWS]). This document is part of MUWS.

MUWS defines how an Information Technology resource connected to a network provides manageability interfaces such that the IT resource can be managed locally and from remote locations using Web services technologies.

MUWS is composed of two parts. This document is MUWS part 1 and provides the fundamental concepts for management using Web services. MUWS part 2 [MUWS Part 2] provides specific messaging formats used to enable the interoperability of MUWS implementations. MUWS part 2 depends on MUWS part 1, while part 1 is independent from part 2.

Status:

This document is a committee draft of version 1.0. There is no guarantee that any part of the content in this document will appear in the final, released MUWS 1.0 specification.

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

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

The errata document for this specification is maintained at:
http://docs.oasis-open.org/wsdm/2004/12/cd-wsdm-muws-part1-1.0-errata.pdf

Table of Contents

1 Introduction

Management Using Web Services (MUWS) enables management of distributed information technology (IT) resources using Web services. Many distributed IT resources use different management interfaces. By leveraging Web service technology, MUWS enables easier and more efficient management of IT resources. This is accomplished by providing a flexible, common framework for manageability interfaces that leverage key features of Web services protocols. Universal management and interoperability across the many and various types of distributed IT resources can be achieved using MUWS.

The types of management capabilities exposed by MUWS are the management capabilities generally expected in systems that manage distributed IT resources. Examples of manageability functions that can be performed via MUWS include:

- monitoring the quality of a service
- enforcing a service level agreement
- controlling a task
- managing a resource lifecycle

MUWS is designed to meet the requirements defined in the MUWS Requirements document [MUWS REQS]. Whenever possible, MUWS leverages existing Web services specifications to ensure interoperability, adoptability, and extensibility.

There is a basic set of manageability capabilities defined in this specification. The only capability required by MUWS is the *Identity* capability defined in section 5.1.

To understand the various topics discussed in this specification, the reader should be familiar with IT management concepts. In addition, the following assumptions are made:

- The reader is familiar with the Web Services Architecture [WSA]**.**
- The reader is familiar with XML [XML 1.0 3rd Edition], XML Schema[XML Schema Part 1] [XML Schema Part 2], and XML Namespace [XNS].
- The reader is familiar with WSDL [WSDL], SOAP [SOAP] and WS-Addressing [WS-Addressing].
- The reader is familiar with WS SOAP Message Security [WSS].

The text of this specification, along with Appendix C and Appendix D, is normative with the following exception: the abstract, examples, UML diagrams and any section explicitly marked as non-normative.

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 RFC 2119 [RFC2119].

Furthermore, this specification defines and uses the following terms:

Web service endpoint – an entity providing a destination for Web service messages. A Web service endpoint has an address (URI) and is described by the content of a WSDL 1.1 port element. This definition is consistent with the definition provided in the WS-Addressing specification [WS-Addressing].

Web service interface – a group of operations described by the content of a WSDL 1.1 portType element. These operations can provide access to resource properties and metadata.

Resource –a logical or physical component of some subject domain, for example, a printer, a magnetic storage disk, an application server, a CRM application or a car engine.

Manageable resource – a resource capable of supporting one or more standard manageability capabilities.

Capability –a group of properties, operations, events and metadata, associated with identifiable semantics and information and exhibiting specific behaviors.

Manageability – the ability to manage a resource, or the ability of a resource to be managed.

Manageability capability – a capability associated with one or more management domains.

Standard manageability capability – a manageability capability that is defined by this specification.

Manageability interface –the composition of one or more manageability capability interfaces.

Manageability capability interface –a Web service interface representing one manageability capability.

Manageability consumer –a user of manageability capabilities associated with one or more manageable resources.

Manageability endpoint –a Web service endpoint associated with and providing access to a manageable resource.

Management domain – an area of knowledge relative to providing control over, and information about, the behavior, health, lifecycle, etc. of manageable resources.

1.2 Notational conventions

This specification uses an informal syntax to describe the XML grammar of the information used in defining the management capability interfaces. This syntax uses the following rules:

- The syntax appears as an XML instance, but data types appear instead of values.
- {any} is a placeholder for elements from some other namespace (like ##other in the XML Schema).
- The Cardinality of an attribute, element, or {any}, is indicated by appending characters to the item as follows:

?	none, or one
*	none, or more
+	one, or more
no character	exactly one

- Items contained within the square brackets, [and], are treated as a group.
- Items separated by | and grouped within parentheses, (and), indicate syntactic alternatives.
- An ellipsis, or three consecutive periods, ..., are used in XML start elements to indicate that attributes from some other namespace are allowed.
- The XML namespace prefixes, defined in section 4, indicate the namespace of an attribute or an element.

A full XML Schema description of the XML information is available in Appendix C of this specification.

When describing an instance of XML information, and in order to refer to an element or an attribute, this specification uses a simplified Xpath-like notation that is formally defined as follows:

Path = '/'? (['@'? (NCName | QName | '*')] | ['(' (NCName | QName | '*'] ')') ['/' Path]?

where:

- *NCName* is an XML non-qualified name as defined by the 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 defines by the XML Schema [XMLS].
- Symbol * denotes any name match.
- Symbol / denotes a path delimiter. When it appears as the first element of the path, it denotes the root of the XML document.
- Symbol @ denotes a reference to an XML attribute. If absent then an NCName, QName or * refer to an XML element.
- 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 element E2 which is contained in 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 element E1, as declared in the XML Schema type T1. In this case, the target namespace, T1, is mapped to the prefix ns2.

2 Architecture

This WSDM specification (MUWS) defines how the ability to manage, or, how the *manageability of*, an arbitrary *resource* can be made accessible via *Web services*. In order to achieve this goal, MUWS is based on a number of Web services specifications, mainly for messaging, description, discovery, accessing properties, and notifications (section 3). Some of these Web services specifications are first presented in [MUWS Part 2].

The basic concepts of management using Web services can be illustrated by the following figure:

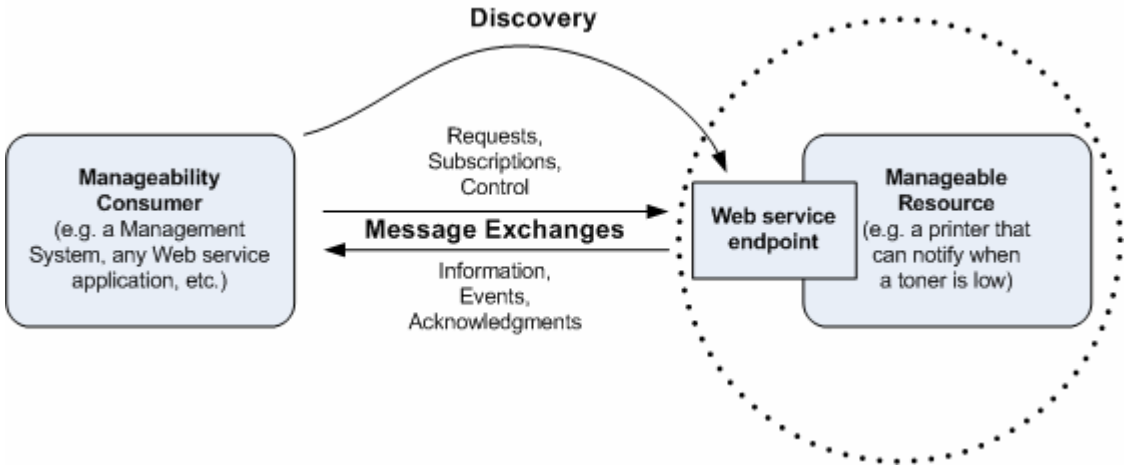

Figure 1: WSDM Concepts

A Web service *endpoint* provides access to a *manageable resource*. An example of a manageable resource is a printer that indicates when its toner is low, or, a magnetic storage disk that reports its internal temperature,

A *manageability consumer* discovers the Web service endpoint and *exchanges messages* with the endpoint in order to request information, subscribe to events, or, control the manageable resource associated with the endpoint. An example of a manageability consumer is a management system, or, a business automation process, or simply, any Web service application.

In order to discover the Web service endpoint providing access to a particular manageable resource, a manageability consumer first obtains an Endpoint Reference (EPR), as defined by the WS-Addressing specification [WS-Addressing], and then obtains any other required descriptions, including, but not limited to, a WSDL document [WSDL], an XML Schema, or a policy document. MUWS uses the same mechanisms, for obtaining EPRs and their associated descriptions, as used by regular Web service implementations, and their applications.

A Web service endpoint providing access to some manageable resource is called a *manageability endpoint*.

To exchange messages with a manageability endpoint, a manageability consumer needs to understand all of the required descriptions for the endpoint. The manageability consumer sends messages targeted to the manageable resource by using information contained in the EPR, for example, an address and some reference properties (see [WS-Addressing]).

2.1 Focus on Resources

The WSDM specification focuses upon how access is provided to manageable resources. Essentially, there exists a contract between a manageability consumer and a manageable resource with respect to the ability of the consumer to understand what messages can be exchanged between the consumer and the resource. Therefore, the central element and focal

point of the WSDM architecture is the manageable resource. The message patterns encapsulate access to resources into manageable resources instead of exposing message patterns to indirectly access the resource through agents, proxies, observers, etc.

2.1.1 Capabilities for Management

Manageability is one possible aspect of a resource. For example, a printer can (obviously!) print. Printing is the functional/operational aspect of the printer. However, the same printer may be able to indicate if it is on-line, or, if the toner has run out. Such indications compose manageability capabilities of the printer. A manageable resource may support some number of capabilities. Each capability has distinct semantics, for example, an ability to describe relationships among resources or an ability to indicate if the resource is on-line or off-line. An implementation of a manageable resource provides a set of manageability capabilities via Web service endpoints.

In WSDM terms, a *manageability capability*

- is uniquely identified,
- has defined semantics (such as those provided by any section in this specification that describes a new capability),
- is associated with a set of properties, operations, events (notifications) and metadata (including policies).

Each manageability capability defined in the WSDM specifications is extensible. New capabilities can be similarly defined, based on a particular resource manageability model, for example, DMTF CIM. MUWS provides mechanisms, patterns, and refinements, for defining new manageability capabilities and for discovering, identifying and using capabilities of a manageable resource.

2.1.2 Isolation from Implementation

The WSDM architecture focuses upon the manageable resource. This approach does not restrict choices of an implementation strategy. Moreover, WSDM isolates the manageability consumer from implementation specific aspects of a manageable resource or Web service endpoint. For example, a direct-to-resource, agent-less approach, or, an approach using management agents are equally valid implementations. Such implementation details are transparent to manageability consumers. Figure 2 illustrates this point:

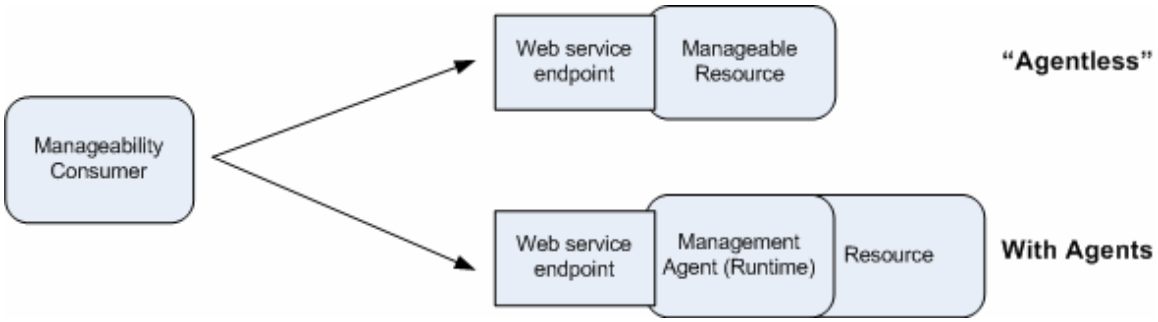

Figure 2: Isolation from Implementation

2.2 Composability

Composability allows a manageable resource's implementation to support a non conflicting mix of some number of capabilities as well as features provided by the Web services platform. Parts of the composition incrementally enrich the implementation without incurring disruptions. For example, a SOAP message sent to a Web service endpoint may result in an order being placed. A similar SOAP message with WS-Security headers, signed and encrypted, may result in an order being placed in a secure manner. The mix of the order placement, plus the security implemented by a Web service endpoint, leveraged message-level composability. In other words,

the SOAP message is composed of an order placement request, plus the appropriate security headers, encryption and digital signatures.

The implementer of a manageable resource may create an appropriate composition of aspects and capabilities offered to a manageability consumer via one or more Web service endpoints. Within the context of WSDM, there are two kinds of composition that can manifest in an implementation of a manageable resource, as follows:

1. ***Composition of aspects of a Web services implementation*** – for example, messaging, description, discovery, security, asynchronous notifications, etc. These implementation aspects are provided by the Web services platform and the respective standards specifications (see section 3).
2. ***Composition of manageability capabilities***, which may be classified into one of two categories, as follows:
    a. ***Composition of common manageability capabilities*** – for example, the ability to identify manageable resources, the ability to report and notify on a change of resource availability, or, the ability to report on how resources are related to each other. Such common manageability capabilities are defined in this specification in section 4 and in [MUWS Part 2]. Essentially these are base-line enablers of a richer set of resource manageability. This is similar to how SOAP and HTTP may be considered baseline enablers of Web services.
    b. ***Composition of resource-specific manageability capabilities*** – for example, an ability to manage printers, or, an ability to manage network-connected devices. Other specifications define these manageability capabilities based on the available resource management model, (e.g. DMTF CIM), based on the needs of the management applications, based on the abilities of the resource (e.g. WSDM MOWS), or based on the needs of the management application.

The whole composition as implemented by a manageable resource is then accessible via a Web service endpoint. This is illustrated in Figure 3.

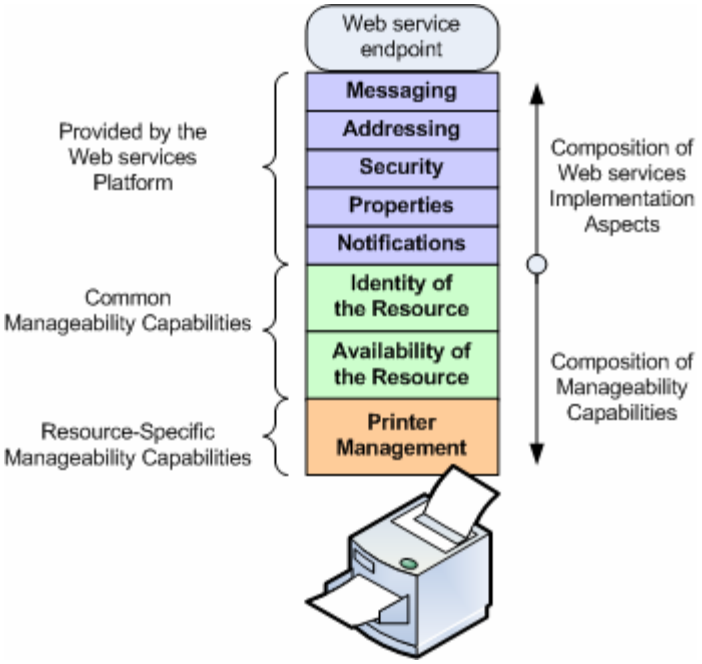

Figure 3: Composability

2.2.1 Low-end to High-end Manageability

The WSDM architecture provides appropriate coverage from low-end manageability of small devices like mobile phones, to high-end manageability of very capable components like application servers and business processes. This range of coverage is achieved by the low barrier to entry placed upon a WSDM implementation: there are few normative requirements made by this specification and the specifications it depends on. Also, composability allows for additional manageability capabilities to be gradually introduced, based upon the availability of management functions and processing power within an implementation of a manageable resource. Manageability consumers can discover and make use of composed capabilities as these capabilities become available. This flexibility is built into the foundation of the WSDM architecture (Figure 4).

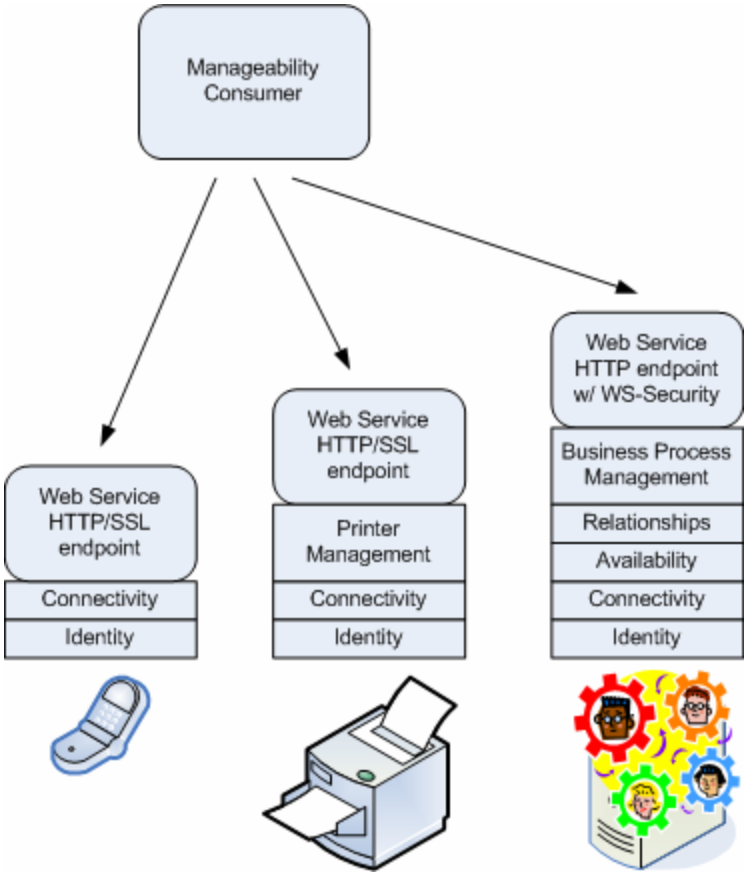

Figure 4: Low-end to High-end Manageability

2.3 Formal Representation of the Architecture

The following UML 2.0 model captures the WSDM MUWS concepts within the context of the WSDL 1.1 [WSDL] and WS-Addressing [WS-Addressing] specifications. Figure 5 provides a "mind map", or digest of the concepts described within the WSDM Architecture.

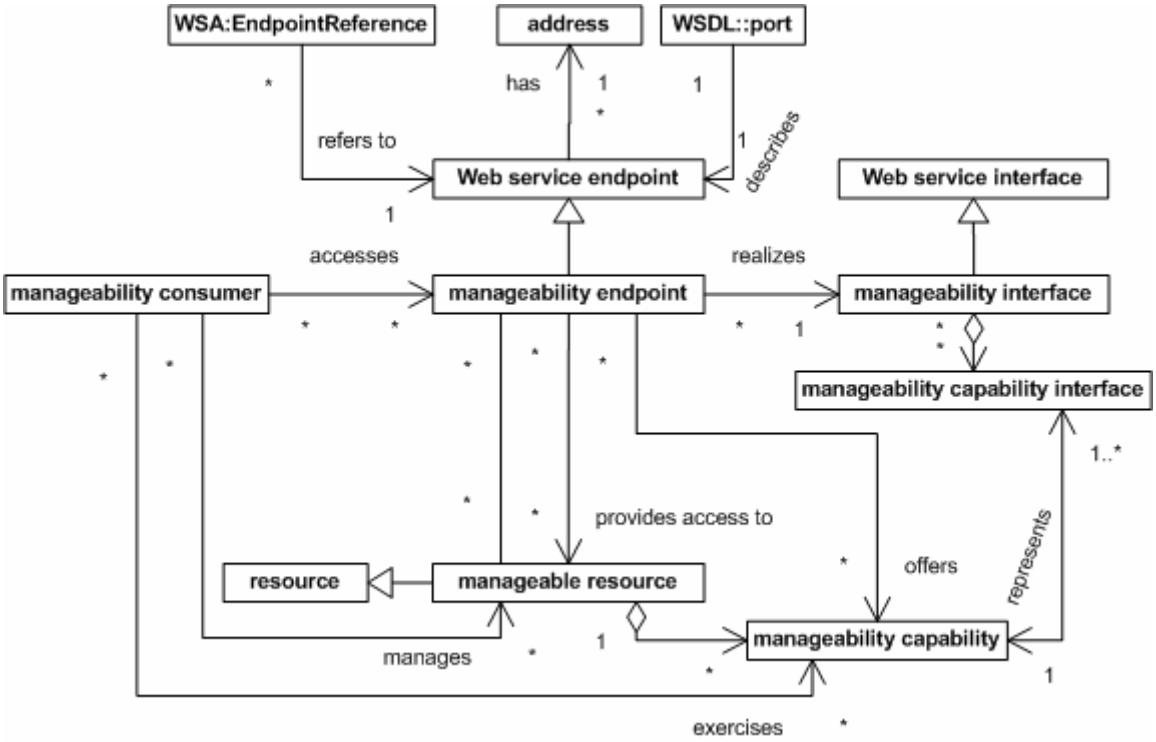

Figure 5: Formal expression of the WSDM architecture concepts

3 Usage of the Web Services Platform

As described in section 2, the foundation for MUWS is provided by the Web services platform. A number of Web services specifications may be composed with the WSDM specifications when implementing a manageability endpoint for a manageable resource. This and dependent specifications are used to represent different aspects of a capability: the properties, the operations, metadata, and events. [MUWS Part 2] introduces additional Web services specifications to define an interoperable way to represent these capability aspects.

3.1 Properties

MUWS uses XML Schema ([XML Schema Part 1], [XML Schema Part 2]) to describe properties. A MUWS property is represented by a Global Element Declaration (GED). In order to create a property one MUST provide:

- the schema for the property,
- a description (in some form) of the semantics of the property,
- the cardinality of the property,
- any relevant metadata for the property.

A manageable resource MUST expose an XML document containing, as top-level elements, all the properties of the manageable resource. This document is called the resource properties document for the resource.

3.2 Operations

MUWS uses [WSDL] to describe operations. The “operations” component of a capability corresponds to an operation, as defined by WSDL. In order to create an operation one MUST provide:

- a WSDL portType containing a WSDL operation corresponding to the operation,
- a description (in some form) of the semantics of the operation,
- any relevant metadata for the operation.

3.3 Events

Event types (as opposed to instances of event messages) are defined in MUWS by providing the combination of a “topic” QName and a “message content” Global Element Declaration. The “topic” QName need not be the QName of the “message content” element. A “topic” or a “message content” element need not be exclusive to one event. However, the combination of a “topic” and a “message content” element MUST uniquely identify an event. The “message content” element represents information that is transmitted as part of a notification message and corresponds to an event instance. The “topic” provides information about why the event was generated. In order to create a new event, one MUST provide:

- the corresponding “topic” and ”message content” element,
- a description (in some form) of the semantics for the “topic” and ”message content” element,
- any relevant metadata for the event.

A manageability endpoint SHOULD offer one or more events that correspond to a change in the properties it supports.

3.4 Metadata

MUWS allows definition of metadata on properties and operations. One such metadata item on properties is whether it is *Mutable*. Mutability is defined as an indication of whether the value of a property can change over time. Another metadata item on a property is whether it is *Modifiable*. Modifiability is defined as an indication of whether the value of a property can be set explicitly, as opposed to can not be set at all, or, can be set only as a side-effect of setting some other property. Finally, a *Capability* is a metadata item that can be attached to a property, an operation or an event. This metadata item contains a unique identifier for the capability. [MUWS Part 2] describes additional metadata items.

For each property introduced in this specification, the value of these metadata items is described. However, MUWS does not specify if, or how, the value is made available to a consumer.

3.5 Addressing

MUWS makes use of the endpoint reference (EPR) construct, as defined in [WS-Addressing]. In addition, MUWS-compliant messages MUST comply with the rules in [WS-Addressing] regarding the use of SOAP headers, and, regarding how the content of the EPR constrains the messages sent to the endpoint.

3.6 Security

When evaluating the security requirements for resource management, it is important to delineate several aspects of Security technology;

- Identification: Presentation of a claimed identity
- Authentication: Verification of proof of asserted identity
- Authorization: The information and mechanisms to allow appropriate authorized requests to resources and deny unauthorized requests.
- Message Integrity: The protection of messages in a message exchange from unauthorized modification.
- Data Integrity: The protection of data from unauthorized modification.
- Data confidentiality
- Trust

A complete security model addressing the requirements listed above needs to be provided for any management deployment. Profiles for different sets of requirements will be needed to ensure interoperable deployments.

An explicit mapping to an authorization model at deployment time should be provided by a conformant management application.

To address security of messages, MUWS relies on generic Web services security mechanisms, including transport-level security (e.g. HTTP over SSL), OASIS Web Services Security message-level security [WSS], etc. The composition of appropriate security specifications and this specification provides a model for securing the messages exchanged during management using Web services realized by manageability endpoint implementations. The choice of concrete security mechanisms should be carried out by the implementers of the manageability endpoints and may conform to some profile.

Within an enterprise MUWS can be deployed like any other specification into the existing enterprise security model. When managing between enterprises, security will need to be developed in an ad hoc, pair-wise fashion at a messaging level.

This specification defines some metadata items for management. Whenever information related to management metadata is being relied on, it is important to understand the environment in which the metadata is being asserted. It may be needed to provide some data integrity mechanisms to protect the information from unauthorized modification. It may also be needed to

406 implement a set of authorization mechanisms to provide a way of identifying under what
407 conditions information should be shared.

4 Common Information Items

4.1 WSDM Event Format

The WSDM Event Format defines an XML format to carry management event information. The format defines a set of basic, consistent data elements that allow different types of management event information to be carried in a consistent manner. The WSDM Event Format provides a basis for programmatic processing, correlation, and interpretation of events from different products, platforms, and management technologies.

The WSDM Event Format organizes management event data into three basic categories, the event reporter, the event source, and extensible, event-specific, situation data. Each category contains a few common properties, as found in most management events, and allows for extensible, event-specific data. The WSDM Event Format has a flexible and extensible syntax..

To be effective, the WSDM Event Format MUST provides the following essential information:

- the identification of the resource experiencing an event, called the source,
- the identification of the reporter of an event, known as the reporter. In most cases the source reports its own event, thus the identity of the reporter and the source are the same.

Typically, further information is also needed to describe the semantics of an event.

Additionally, an event MUST contain an *EventId* that is unique across event types within the source. An event may contain additional information related to the situation that has occurred or to the context within which it occurred. For example, message text, severity information or related Application Response Measurement (ARM) instrumentation information. It is RECOMMENDED that a container be used to encapsulate additional information that is significant to an event.

The base element of the WSDM Event Format is *muws-p1-xs:ManagementEvent*, as presented in the next section.

4.1.1 XML Representation of the event

The following is the XML representation of the WSDM MUWS management event container.

```
<muws-p1-xs:ManagementEvent ...
  muws-p1-xs:ReportTime="xs:dateTime"?>

  <muws-p1-xs:EventId>xs:anyURI</muws-p1-xs:EventId>

  <muws-p1-xs:SourceComponent ...>
    <muws-p1-xs:ResourceId>xs:anyURI</muws-p1-xs:ResourceId> ?
    <muws-p1-xs:ComponentAddress>{any}</muws-p1-xs:ComponentAddress> *
    {any}*
  </muws-p1-xs:SourceComponent>

  <muws-p1-xs:ReporterComponent ...>
    <muws-p1-xs:ResourceID>xs:anyURI</muws-p1-xs:ResourceId> ?
    <muws-p1-xs:ComponentAddress>{any}</muws-p1-xs:ComponentAddress> *
    {any}*
  </muws-p1-xs:ReporterComponent> ?
  {any}*
</muws-p1-xs:ManagementEvent>
```

Where the clauses are described as follows:

muws-p1-xs:ManagementEvent: The wrapper element used for management event messages.

muws-p1-xs:ManagementEvent/@muws-p1-xs:ReportTime: The date and time when the event was reported. If the value does not include a time zone designation, or use 'Z' for UTC, then the value MUST be interpreted as having a time zone of UTC. The value of *ReportTime* MUST provide a granularity as precise as is supported by the generating platform. This attribute is RECOMMENDED.

muws-p1-xs:ManagementEvent/muws-p1-xs:EventId: The primary identifier for an event. This element MUST be unique within the scope provided by the manageability implementation for the source resource. This element MAY be used as the primary key for the event. This element is provided for management functions that require events to have an identifier. It is of type URI and is REQUIRED.

muws-p1-xs:ManagementEvent/muws-p1-xs:SourceComponent: The identification of, or reference to, the source associated with an event. This element is REQUIRED.

muws-p1-xs:ManagementEvent/muws-p1-xs:SourceComponent/ResourceId: A specification of an identifier of a manageable resource associated with an event. This is an OPTIONAL property. This property is intended as an identifier to be used, for example, in correlation, so that management consumers can ensure that information contained in the *muws-p1-xs:ManagementEvent* pertains to a given manageable resource. If provided, this element MUST correspond to the *muws-p1-xs:ResourceId* property (defined in section 5.1.2) for the source associated with an event.

muws-p1-xs:ManagementEvent/muws-p1-xs:SourceComponent/muws-p1-xs:ComponentAddress: Contains the specific elements used to identify the address of a component. If this element contains more than one child element, each child element represents an alternate address of the same source. This element is RECOMMENDED to improve interoperability.

muws-p1-xs:ManagementEvent/muws-p1-xs:SourceComponent/muws-p1-xs:ComponentAddress/{any}: XML open content including any XML representation of the component address. One commonly used address type is a Web service address, such as an EPR as defined by [WS-Addressing]. In the case where the source is a manageable resource, it is RECOMMENDED that the *muws-p1-xs:ManageabilityEndpointReference* element, as defined in section 4.2, be used as the address type.

muws-p1-xs:ManagementEvent/muws-p1-xs:ReporterComponent: Provides the identification of, or reference to, the reporter associated with an event. This is a REQUIRED property only if the reporter is different from the source. Otherwise, this element is OPTIONAL. When this element is absent the reporter is asserted to be the same as the source. The content of this element is the same as the content of the *ManagementEvent/SourceComponent* element except that the definitions apply to the reporter rather than the source.

muws-p1-xs:ManagementEvent/{any}: XML open content providing a container for additional data associated with an event. Among other things, this is where the "message content" Global Element Declaration introduced in section 3.3 is inserted. MUWS Part 2 defines some additional element that could be included using this wildcard.

4.2 Manageability Endpoint Reference

MUWS defines the following element to represent a reference to a manageability endpoint:

```
<muws-p1-xs:ManageabilityEndpointReference>
  wsa:EndpointReferenceType
</muws-p1-xs:ManageabilityEndpointReference>
```

The element is an EPR as defined by [WS-Addressing]. The EPR provides a reference to a manageability endpoint.

5 Capabilities

There is a minimum set of manageability capabilities that an implementation of a manageability endpoint must support in order to comply with the MUWS specification.

A manageability capability defines properties, operations and events to support domain-specific tasks. Details of a manageability capability are exposed by a manageable resource.

A manageable resource MAY also define a new resource-specific manageability capability.

A manageable resource SHOULD extend a MUWS manageability capability with a resource-specific manageability capability that uses similar semantics. A manageable resource is not required to extend a MUWS manageability capability when a resource-specific manageability capability uses different semantics than the set of MUWS manageability capabilities.

In this section the following namespaces are used unless otherwise specified. The table below lists each prefix and a corresponding namespace URI.

Prefix	Namespace
muws-p1-xs	http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-part1.xsd
pbm	http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-pbm.xsd
xs	http://www.w3.org/2001/XMLSchema
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing

5.1 Identity

The manageability capability URI for the *Identity* capability is
http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Identity

5.1.1 Definition

The goal of the Identity capability is to establish whether two entities are the same. This is a required capability and it MUST be provided by every manageability endpoint. Observe that this requirement does not preclude the manageability endpoint from applying a security policy preventing some requesters from accessing this, or another, capability.

In addition, this capability is used as a “marker” interface enabling a manageability consumer to learn if an endpoint is a manageability endpoint.

Figure 6 shows the UML representation of MUWS Identity.

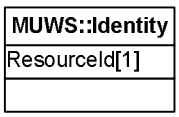

Figure 6: MUWS Identity

5.1.2 Properties

The following is the specification of the property defined by the Identity capability.

```
<muws-p1-xs:ResourceId>xs:anyURI</muws-p1-xs:ResourceId>
```

The following is an example property instance for the property defined by the Identity capability.

```
<muws-p1-xs:ResourceId>
  http://example.com/resource/diskDrive/9F34AD35B
</muws-p1-xs:ResourceId>
```

Note that *ResourceId* is an opaque identifier of a resource managed through a manageability endpoint. *ResourceId* is a read-only, mandatory property with a cardinality of 1.

This property has the following metadata:

It is not *Mutable*.
It is not *Modifiable.*
Its *Capability* is “http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Identity”.

The following constraints are applicable to *ResourceId*:

- Globally unique: A manageability endpoint MUST create the *ResourceId* URI in a way that ensures that the *ResourceId* is unique to the resource managed through the manageability endpoint and globally unique. This specification does not prescribe the means by which global uniqueness is achieved.
- Uniqueness in time: A *ResourceId* MUST NOT be reused by the implementation of a manageability endpoint for another resource, even after the original resource no longer exists.
- Consistency across endpoints: An implementation of a manageability endpoint SHOULD use a *ResourceId* that is suggested by the characteristics of a resource. This is possible when, for example, a *ResourceId* is retrievable from a resource by a manageability endpoint, or, an application of MUWS to a given domain specifies a method for building a *ResourceId* based upon characteristics of resources populating the domain. It is not guaranteed that different manageability endpoints associated with the same resource will, in all cases, return the same *ResourceId*.
- Consistency within an endpoint: An implementation that exposes several manageability endpoints for the same resource MUST report the same *ResourceId* at each manageability endpoint.
- Persistence: A manageability endpoint SHOULD return the same *ResourceId* during the entire lifetime of the manageability endpoint, including across power cycles of the manageability endpoint. Resources that are not able to persist a *ResourceId* across power cycles of a manageability endpoint SHOULD try to provide a consistent *ResourceId* via predictable identifier generation or delegation of identity assignment.
- Equality: If two reported *ResourceIds* are equal, then the consumer knows that the two manageability endpoints represent the same resource. The converse proposition is not necessarily true: two different *ResourceIds* could conceivably correspond to the same resource. It is strongly RECOMMENDED that this condition be avoided in a conscious and deliberate manner, as some managers may not be able to distinguish that two different reported identifiers are, in fact, associated with the same manageable resource. Thus, manageability consumers would be forced to treat every identifier as corresponding to a unique manageable resource.

Note that a manageability consumer MUST NOT assume that two manageability endpoints represent two different resources solely because the two reported *ResourceIds* are different.

Since the *ResourceId* is defined as opaque, this specification does not allow a consumer to infer any characteristic of a resource by examining a *ResourceId*, other than comparing the *ResourceId* to another *ResourceId* as one way of establishing oneness. For example, one possible way to construct a *ResourceId* and ensure its uniqueness is to use a UUID wrapped in a URI.

Note that this specification does not define equivalence of URIs and the consumer should decide which level of the comparison ladder, as defined in section 6 of [RFC2396bis], is appropriate to use for this comparison.

MUWS defines an additional mechanism for establishing oneness of two resources. This mechanism, called *Correlatable Properties* is defined in the section 5.3.

5.2 Manageability Characteristics

The manageability capability URI for the *Manageability Characteristics* capability is http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/ManageabilityCharacteristics

5.2.1 Definition

The Manageability Characteristics capability defines properties providing information about the characteristics of a manageability endpoint implementation rather than the resource.

Figure 7 shows the UML representation of MUWS Manageability Characteristics.

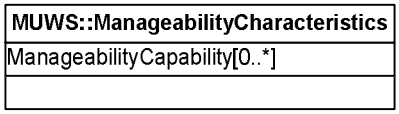

Figure 7: MUWS Manageability Characteristics

5.2.2 Properties

The following is the specification of the property defined by the Manageability Characteristics capability.

```
<muws-p1-xs:ManageabilityCapability>
  xs:anyURI
</muws-p1-xs:ManageabilityCapability>*
```

The following are example of property instances for the property defined by the *Manageability Characteristics* capability.

```
<muws-p1-xs:ManageabilityCapability>
  http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Identity
</muws-p1-xs:ManageabilityCapability>
<muws-p1-xs:ManageabilityCapability>
  http://example.com/capabilities/FooCapability
</muws-p1-xs:ManageabilityCapability>
```

Note that **ManageabilityCapability** contains a URI identifying a manageability capability that is supported by a manageable resource. The cardinality of this property is zero to unbounded.

This property has the following metadata:

It is not *Mutable*.
It is not *Modifiable*.
Its *Capability* is “http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/ManageabilityCharacteristics”.

A manageability interface is said to provide a capability if it supports all of the required properties, events, operations and metadata defined by the capability. This does not preclude the manageability endpoint from applying a security policy preventing some requesters from accessing this, or another, capability.

There SHOULD be one *ManageabilityCapability* property instance for each manageability capability provided by a manageability interface. For capabilities extending a base capability, both the extension and the base capability MUST be listed. Marking a property, operation or event as part of a capability is considered a hint for the consumer of a manageability endpoint. The meaning of such a hint is defined by the capability. As a result, the *ManageabilityCapability*

property facilitates discovery and introspection by providing a hint to the manageability consumer about what requests can be sent to the manageability endpoint.

5.3 Correlatable Properties

The manageability capability URI for the *Correlatable Properties* capability is http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/CorrelatableProperties

5.3.1 Definition

The *Correlatable Properties* capability allows a manageability endpoint to expose its understanding of which property values could be compared when establishing that the manageability endpoint in question and another manageability endpoint correspond to the same resource. This is especially useful in the case where the two manageability endpoints are unable to return the same *ResourceId* for a resource. For example, one manageability endpoint may enable a temperature control capability for a SCSI hard disk drive, and another manageability endpoint may enable a capacity management capability for the same SCSI hard disk drive. Each manageability endpoint may return its own unique *ResourceId* due to implementation requirements or constraints (e.g. firmware). However, implementers of a manageability endpoint may be aware of some unique resource-specific property values that can indicate if two manageability endpoints correspond to the same resource. In the SCSI example, correlatable properties could be host IP, bus #, channel #, SCSI ID, LUN ID. If the values of those property instances match, then one could be fairly certain that multiple manageability paths are provided to the same SCSI resource.

Using the CorrelatableProperties capability, both manageability endpoints may expose their understanding of what resource property values need to match in order to establish a correlation between manageable resources. The manageability consumer uses this information to evaluate and establish such a correlation.

Note that if the *ResourceIds* returned by both manageability endpoints are the same but the correlatable properties do not match, then the resources should be considered the same, as the Identity capability takes precedence over *Correlatable Properties* capability. Typically, manageability consumers will not evaluate correlatable properties if the two manageability endpoints return the same *ResourceId*.

Figure 8 shows the UML representation of MUWS *Correlatable Properties*.

Figure 8: MUWS Correlatable Properties

The exposure of the information provided as part of this capability allows clients to understand the information used to uniquely identify the resource. This may allow a nefarious client to spoof the presence of the resource. This is particularly true if it is obvious how to generate or construct the *ResourceId* from these properties. These properties should be used and exposed with this risk in mind. The *CorrelateableProperties* property should receive the same level of protection as the *ResourceID*.

5.3.2 Information Markup Declarations

There are three elements, as defined by this specification, providing a simple property boolean match (PBM) dialect that can be used to express a correlation condition for correlatable properties. This condition is expressed based on values of properties of the two resources that are compared through the correlatable properties mechanism. These elements are defined in a separate namespace, from the rest of the MUWS specification, as follows:

```
<pbm:Match>xs:QName</pbm:Match>
```

This element evaluates to true if the values of the properties for the given QName match for the two resources.

```
<pbm:MatchAny>(<pbm:Match/>|<pbm:MatchAll>)*</pbm:MatchAny>
```

This element evaluates to true if any of the enclosed *Match* and/or *MatchAll* conditions evaluate to true.

```
<pbm:MatchAll>(<pbm:Match/>|</pbm:MatchAny>)*</pbm:MatchAll>
```

This element evaluates to true if all of the enclosed *Match* and/or *MatchAny* conditions evaluate to true.

5.3.3 Properties

The following is a definition of the property defined by the *Correlatable Properties* capability.

```
<muws-p1-xs:CorrelatableProperties
  Dialect="xs:anyURI"
  NegativeAssertionPossible="xs:boolean"?>
   {any} *
</muws-p1-xs:CorrelatableProperties>*
```

This property indicates, from the perspective of the manageability representation, which property values, conditions and expressions are used to correlate a manageable resource. The cardinality of the property is zero to unbounded.

This property has the following metadata:

It is *Mutable*.
It is not *Modifiable*.
Its *Capability* is "http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/CorrelatableProperties".

The value of this property is the correlation expression. The format of the correlation expression is determined by the *Dialect* attribute. This specification defines three possible dialect values. An additional dialect value can be defined to provide additional functionality. A manageability representation can offer several instances of the *muws-p1-xs:CorrelatableProperties* property, using the same, or different, dialects. A manageability consumer may evaluate a *muws-p1-xs: CorrelatableProperties* property in any dialect that it understands. Support for a particular dialect is optional.

The dialects defined by this specification are:

- Simple Property Boolean Match
  The URI for this dialect is http://docs.oasis-open.org/wsdm/2004/12/pbm.
  The content of the property is as described in section 5.3.2. If all top-level match conditions evaluate to true, then a correlation between manageable resources is established.
- XPath 1.0
  The URI for this dialect is http://www.w3.org/TR/1999/REC-xpath-19991116.
  The content of the property is an [XPath 1.0] expression. When retrieved as a property form a manageable resource, the XPath expression is evaluated on properties of another manageability resource. If the XPath expression evaluates to a Boolean value of *true*, or if it evaluates to a non-empty, non-boolean value, without any errors, then a correlation is established between the manageable resources.
- XPath 2.0
  The URI for this dialect is http://www.w3.org/TR/xpath20/.
  The content of the property is an [XPath 2.0] expression. This XPath expression is

evaluated on a resource properties document of another manageability representation. If the XPath expression evaluates to a Boolean value of *true*, or if it evaluates to a non-empty, non-boolean value, without any errors, then a correlation is established between the manageable resources.

The optional *NegativeAssertionPossible* attributes express whether a negative result from the evaluation of the correlation expression implies that the resources are necessarily different. The default value is false.

- If *NegativeAssertionPossible* is *false*, only a positive match is meaningful to the consumer. In other words, if the correlation expression evaluates successfully, according to the evaluation rules defined by the dialect, then a consumer can consider the resource representations to represent the same resource. If the correlation expression does not evaluate successfully, then the consumer can not infer whether the resource representations represent different resources.
- If *NegativeAssertionPossible* is *true*, a positive match still means that the resources are the same. But a negative match now means that the resources are guaranteed to NOT be the same.

5.3.3.1 Examples of use

Consider the following two simplified sets of properties, obtained through two different manageability endpoints:

Properties obtained through manageability endpoint ME1:

```
<print:PrinterResourcePropDoc>
  ...
  <print:PrinterModel>PrintCo SuperJet 5000</print:PrinterModel>
  <print:Location>Building 42 lower pillar D4</print:Location>
  <print:Owner>Sir Printalot</print:Owner>
  <print:IPAddress>15.244.62.41</print:IPAddress>
  <foo:Name>Baby got ink</foo:Name>
  <muws-p1-xs:CorrelatableProperties
    Dialect="http://docs.oasis-open.org/wsdm/2004/12/pbm">
    <pbm:MatchAny>
      <pbm:Match>print:IPAddress</pbm:Match>
      <pbm:MatchAll>
        <pbm:Match>foo:Name</pbm:Match>
        <pbm:Match>print:PrinterModel</pbm:Match>
        <pbm:Match>print:Location</pbm:Match>
        <pbm:Match>print:Owner</pbm:Match>
      </pbm:MatchAll>
    </pbm:MatchAny>
  </muw-p1-xs:CorreletableProperties>
</print:PrinterResourcePropDoc>
```

Properties obtained through manageability endpoint ME2:

```
<print:PrinterResourcePropDoc>
  ...
  <print:PrinterModel>PrintCo UltraJet 40</print:PrinterModel>
  <print:Location>Building 42 lower pillar D4</print:Location>
  <print:Owner>Sir Printalot</print:Owner>
  <print:IPAddress>15.244.10.89</print:IPAddress>
  <foo:Name>Baby got ink</foo:Name>
</print:PrinterResourcePropDoc>
```

The *CorrelatableProperties* property, as provided through manageability endpoint ME1, asserts that if a manageability representation provides a view of a resource which either has the same *IPAddress* as ME1, or, has the same *Name*, *PrinterModel*, *Location*, and *Owner* as ME1, then these two manageability endpoints represent are the same printer. In this example, since the

IPAddress doesn't match and the *PrinterModel* is different, the correlation is not established and the consumer cannot deduce that the two printers are the same.

Note that since the *NegativeAssertionPossible* attribute is not specified on *CorrelatableProperties* it takes the default value of *false*. Therefore, the consumer cannot assume that the resources are indeed two different printers. At this point, the consumer still cannot infer whether the two manageability endpoints correspond to the same printer or not.

Properties obtained through manageability endpoint ME3:

```
<print:PrinterResourcePropDoc>
  ...
  <muws-p1-xs:CorrelatableProperties
    Dialect=http://www.w3.org/TR/1999/REC-xpath-19991116
    NegativeAssertionPossible=”false”>
boolean(/print:PrinterResourcePropDoc/print:LastJob/print:JobID="5622654845
1262") and
boolean(/print:PrinterResourcePropDoc/print:LastJob/print:JobOriginator="15
.244.30.30")
  </muw-p1-xs:CorrelatableProperties>
</print:PrinterResourcePropDoc>
```

Properties obtained through manageability endpoint ME4:

```
<print:PrinterResourcePropDoc>
  ...
  <print:LastJob>
    <print:JobID>56226548451262</print:JobID>
    <print:JobOriginator>15.244.30.30</print:JobOriginator>
    <print:JobDate>2004-03-11T11:30:56Z</print:JobDate>
  </print:LastJob>
</print:PrinterResourcePropDoc>
```

The *CorrelatableProperties* property, as provided through manageability endpoint ME3, asserts that if a manageability endpoint provides a view of a resource for which the *JobID* of the last job is 56226548451262, and the *JobOriginator* of the last job is 15.244.30.30, then these manageability endpoints represent the same printer. In this example, the condition is satisfied, so the consumer knows that ME3 and ME4 correspond to the same physical printer. Note that, as the example shows, with this dialect the consumer only needs to retrieve the *CorrelatableProperties* property and no other property from ME3 to check correlation. From ME4 it needs to retrieve the properties needed to evaluate the XPath expression. In this example, *NegativeAssertionPossible* is set to *false*, thus a negative result would not have guaranteed that the printers behind ME3 and ME4 are indeed different.

6 Defining a Manageability Interface

Implementers of manageability endpoints are free to expose additional manageability capabilities beyond those defined in MUWS. An additional capability is represented by a set of manageability capability interfaces. The properties defined in a new capability must be defined as XML Schema Global Element Declarations. The operations defined in a new capability are represented as WSDL 1.1 operations. Furthermore, a manageability endpoint offering a new capability is free to ignore all standard manageability capabilities defined by MUWS except for the *Identity* capability. The MUWS *Identity* capability is REQUIRED.

MUWS-compliant manageability endpoints SHOULD also comply with the WS-I Basic Profile version 1.1 [BP].

7 References

7.1 Normative

[XML1.0 3rd Edition]
Tim Bray, et al., *Extensible Markup Language (XML) 1.0 (Third Edition)*, W3C Recommendation, February 2004, http://www.w3.org/TR/REC-xml

[XML Schema Part 1]
Henry S. Thompson, et al. *XML Schema Part 1: Structures*, W3C Recommendation, May 2001, http://www.w3.org/TR/xmlschema-1/

[XML Schema Part 2]
Paul V. Biron, et al. *XML Schema Part 2: Datatypes*, W3C Recommendation, May 2001, http://www.w3.org/TR/xmlschema-2/

[XNS] Tim Bray, et al., Extensible *Namespaces in XML*, W3C Recommendation, January 1999, http://www.w3.org/TR/REC-xml-names/

[WSDL] Erik Christensen, et al., *Web services Description Language (WSDL) 1.1*, W3C Note, March 2001, http://www.w3.org/TR/wsdl

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

[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.

[RFC2396bis] T. Berners-Lee, et al., *Uniform Resource Identifier (URI): Generic Syntax*, IETF RFC 2396bis-04, February 2004, http://www.ietf.org/internet-drafts/draft-fielding-uri-rfc2396bis-04.txt

7.2 Non-normative

[MOWS] Igor Sedukhin, *Web Services Distributed Management:Management of Web Services (WSDM-MOWS) 1.0*, OASIS Committee Draft, December 2004, http://docs.oasis-open.org/wsdm/2004/12/cd-wsdm-mows-1.0.pdf

[MUWS Part 2] 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

[MUWS REQS] Pankaj Kumar, et al., Requirements – Management Using Web Services, Committee Draft, October 2003, http://www.oasis-open.org/apps/org/workgroup/wsdm/download.php/6185/WSDM-MUWS-Req-committee-draft-1.0-20031002.pdf

[SOAP] Don Box, et al., *Simple Object Access Protocol (SOAP) 1.1*, W3C Note, May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[WSA] David Booth, et al. *Web Services Architecture*, W3C Working Group Note, February 2004, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

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

[BP] Keith Ballinger, et al. *Basic Profile Version 1.1*, WS-I Final Material, August 2004, http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html

Appendix A. Acknowledgements

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.

In addition, the following non-member employees of member companies made contribution to the specification: Maryann Hondo, Ian Springer.

Appendix B. 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 implementers 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 C. MUWS Part 1 Schema (Normative)

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
    targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-muws-
part1.xsd"
    xmlns:muws-p1-xs="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://schemas.xmlsoap.org/ws/2004/08/addressing"

schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>

  <xs:element name="ResourceId" type="xs:anyURI"/>

  <xs:complexType name="IdentityPropertiesType">
    <xs:sequence>
      <xs:element ref="muws-p1-xs:ResourceId"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="IdentityProperties"
              type="muws-p1-xs:IdentityPropertiesType"/>


  <xs:element name="ManageabilityCapability" type="xs:anyURI"/>

  <xs:complexType name="ManageabilityCharacteristicsPropertiesType">
    <xs:sequence>
      <xs:element ref="muws-p1-xs:ManageabilityCapability"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="ManageabilityCharacteristicsProperties"
              type="muws-p1-xs:ManageabilityCharacteristicsPropertiesType"/>


  <xs:complexType name="CorrelatablePropertiesType">
    <xs:sequence>
      <xs:any minOccurs="0" maxOccurs="unbounded"
              namespace="##other" processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="Dialect" type="xs:anyURI"/>
    <xs:attribute name="NegativeAssertionPossible" type="xs:boolean"/>
    <xs:anyAttribute namespace="##other"/>
  </xs:complexType>

  <xs:element name="CorrelatableProperties"
              type="muws-p1-xs:CorrelatablePropertiesType"/>

  <xs:complexType name="CorrelatablePropertiesPropertiesType">
    <xs:sequence>
      <xs:element ref="muws-p1-xs:CorrelatableProperties"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

```

```
    <xs:element name="CorrelatablePropertiesProperties"
                type="muws-p1-xs:CorrelatablePropertiesPropertiesType"/>


    <xs:complexType name="ComponentAddressType">
      <xs:sequence>
        <xs:any namespace="##any" processContents="lax"/>
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="ComponentType">
      <xs:sequence>
        <xs:element name="ResourceId" type="xs:anyURI"
                    minOccurs="0"/>
        <xs:element name="ComponentAddress"
                    type="muws-p1-xs:ComponentAddressType"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:any minOccurs="0" maxOccurs="unbounded"
                namespace="##other" processContents="lax"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##other"/>
    </xs:complexType>

    <xs:complexType name="ManagementEventType">
      <xs:sequence>
        <xs:element name="EventId" type="xs:anyURI"/>
        <xs:element name="SourceComponent" type="muws-p1-xs:ComponentType"/>
        <xs:element name="ReporterComponent" type="muws-p1-xs:ComponentType"
                    minOccurs="0"/>
        <xs:any minOccurs="0" maxOccurs="unbounded"
                namespace="##other" processContents="lax"/>
      </xs:sequence>
      <xs:attribute name="ReportTime" type="xs:dateTime" use="optional"/>
      <xs:anyAttribute namespace="##other"/>
    </xs:complexType>

    <xs:element name="ManagementEvent"
                type="muws-p1-xs:ManagementEventType"/>

    <xs:element name="ManageabilityEndpointReference"
                type="wsa:EndpointReferenceType"/>

  </xs:schema>
```

Appendix D. Properties Boolean Match Schema (Normative)

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
    targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-pbm.xsd"
    xmlns:pbm="http://docs.oasis-open.org/wsdm/2004/12/muws/wsdm-pbm.xsd"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:element name="Match" type="xs:QName"/>

  <xs:complexType name="MatchAllType">
    <xs:choice>
      <xs:element ref="pbm:Match"/>
      <xs:element ref="pbm:MatchAny"/>
    </xs:choice>
  </xs:complexType>

  <xs:complexType name="MatchAnyType">
    <xs:choice>
      <xs:element ref="pbm:Match"/>
      <xs:element ref="pbm:MatchAll"/>
    </xs:choice>
  </xs:complexType>

  <xs:element name="MatchAll" type="pbm:MatchAllType"/>

  <xs:element name="MatchAny" type="pbm:MatchAnyType"/>

</xs:schema>
```