



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

Working Draft, December 2nd 2004

Document identifier:

wd-wsdm-muws-part1-1.0

Location:

<http://docs.oasis-open.org/wsdm/2004/12/wd-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 working 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/wd-wsdm-muws-part1-1.0-errata.pdf>

38 **Table of Contents**

39	1	Introduction.....	3
40	1.1	Terminology	3
41	1.2	Notational conventions	4
42	2	Architecture	6
43	2.1	Focus on Resources.....	6
44	2.1.1	Capabilities for Management	7
45	2.1.2	Isolation from Implementation.....	7
46	2.2	Composability.....	7
47	2.2.1	Low-end to High-end Manageability.....	9
48	2.3	Formal Representation of the Architecture.....	9
49	3	Usage of the Web Services Platform.....	11
50	3.1	Properties.....	11
51	3.2	Operations	11
52	3.3	Events.....	11
53	3.4	Metadata.....	12
54	3.5	Addressing	12
55	3.6	Security.....	12
56	4	Common Information Items.....	14
57	4.1	WSDM Event Format.....	14
58	4.1.1	XML Representation of the event.....	14
59	4.2	Manageability Endpoint Reference	15
60	5	Capabilities.....	16
61	5.1	Identity	16
62	5.1.1	Definition.....	16
63	5.1.2	Properties	16
64	5.2	Manageability Characteristics.....	18
65	5.2.1	Definition.....	18
66	5.2.2	Properties	18
67	5.3	Correlatable Properties.....	19
68	5.3.1	Definition.....	19
69	5.3.2	Information Markup Declarations	19
70	5.3.3	Properties	20
71	6	Defining a Manageability Interface	23
72	7	References.....	24
73	7.1	Normative.....	24
74	7.2	Non-normative.....	24
75	Appendix A.	Acknowledgements	26
76	Appendix B.	Notices.....	27
77	Appendix C.	MUWS Part 1 Schema (Normative).....	28
78	Appendix D.	Properties Boolean Match Schema (Normative)	30
79			

80 1 Introduction

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

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

- 91 • monitoring the quality of a service
- 92 • enforcing a service level agreement
- 93 • controlling a task
- 94 • managing a resource lifecycle

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

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

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

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

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

111 1.1 Terminology

112 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
113 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be
114 interpreted as described in RFC 2119 [RFC2119].

115 Furthermore, this specification defines and uses the following terms:

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

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

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

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

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

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

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

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

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

133 **Manageability capability interface** – a Web service interface representing one manageability
 134 capability.

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

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

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

141 1.2 Notational conventions

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

144

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

151 ?	none, or one
152 *	none, or more
153 +	one, or more
154 no character	exactly one
- 155 § Items contained within the square brackets, [and], are treated as a group.
- 156 § Items separated by | and grouped within parentheses, (and), indicate
 157 syntactic alternatives.
- 158 § An ellipsis, or three consecutive periods, ..., are used in XML start elements
 159 to indicate that attributes from some other namespace are allowed.
- 160 § The XML namespace prefixes, defined in section 4, indicate the namespace
 161 of an attribute or an element.

162

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

165

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

168

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

170

171 where:

172 § *NCName* is an XML non-qualified name as defined by the XML
173 Schema [XMLS]. In this case, the namespace is assumed to default
174 to the namespace of this specification.

175 § *QName* is an XML qualified name as defines by the XML Schema
176 [XMLS].

177 § Symbol * denotes any name match.

178 § Symbol / denotes a path delimiter. When it appears as the first
179 element of the path, it denotes the root of the XML document.

180 § Symbol @ denotes a reference to an XML attribute. If absent then an
181 NCName, QName or * refer to an XML element.

182 § Symbols (and) denote a reference to an XML Schema type.

183

184 For example:

185 /E1/E2/@A1 refers to an attribute, A1, of an element, E2, contained
186 in element E1, which is a root of the XML document.

187

188 E1/ns1:E2/E3 refers to an element, E3, which is contained in element
189 E2 which is contained in element E1, anywhere in the
190 XML document. In this case element E2 belongs to the
191 namespace mapped to the prefix ns1.

192

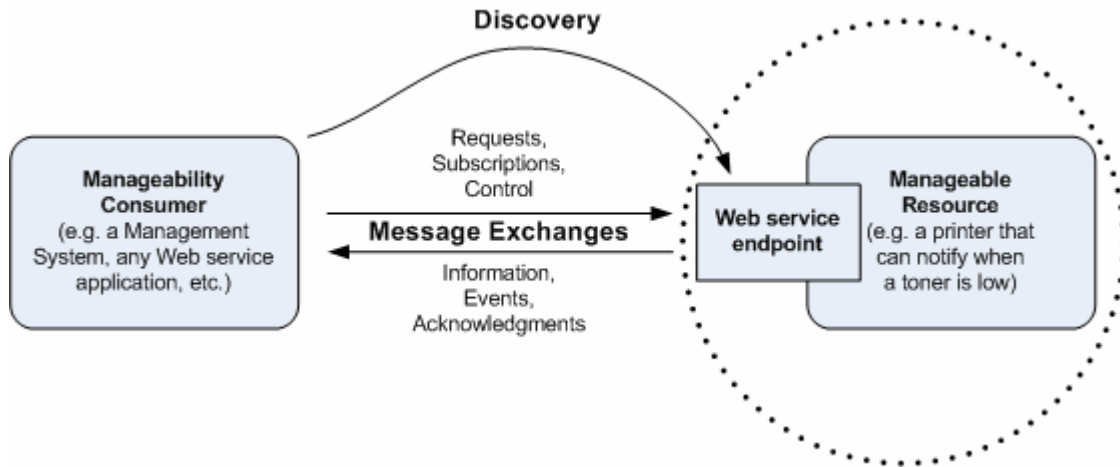
193 (ns2:T1)/E1/ns1:E2/@A1 refers to an attribute, A1, on an element, E2, contained
194 in element E1, as declared in the XML Schema type
195 T1. In this case, the target namespace, T1, is mapped
196 to the prefix ns2.

2 Architecture

197

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

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



204

205

Figure 1: WSDM Concepts

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

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

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

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

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

2.1 Focus on Resources

225

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

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

233 2.1.1 Capabilities for Management

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

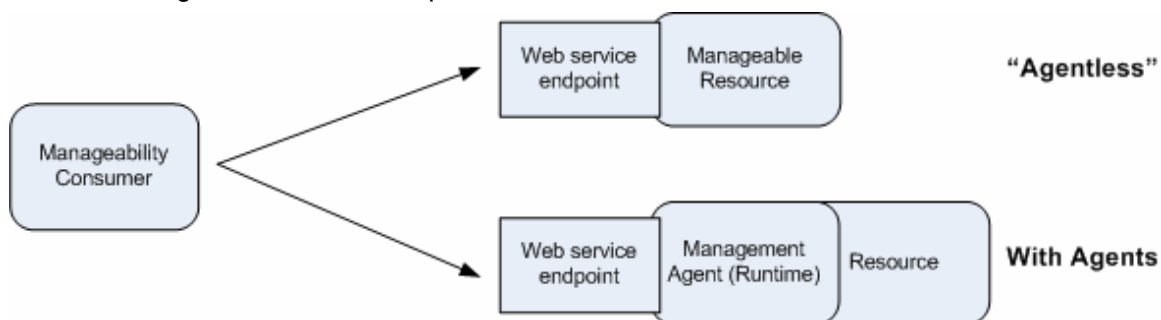
241 In WSDM terms, a *manageability capability*

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

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

251 2.1.2 Isolation from Implementation

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



258
259

Figure 2: Isolation from Implementation

260 2.2 Composability

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

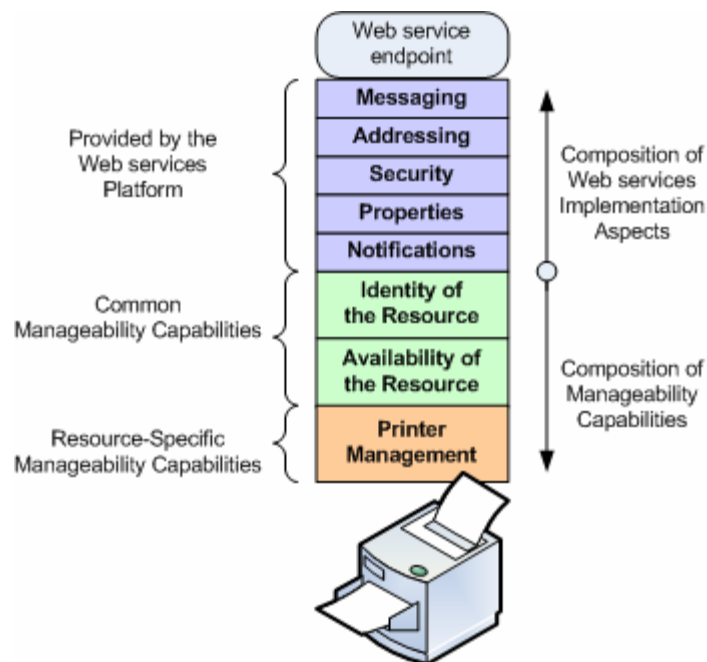
268 the SOAP message is composed of an order placement request, plus the appropriate security
269 headers, encryption and digital signatures.

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

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

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

295



296

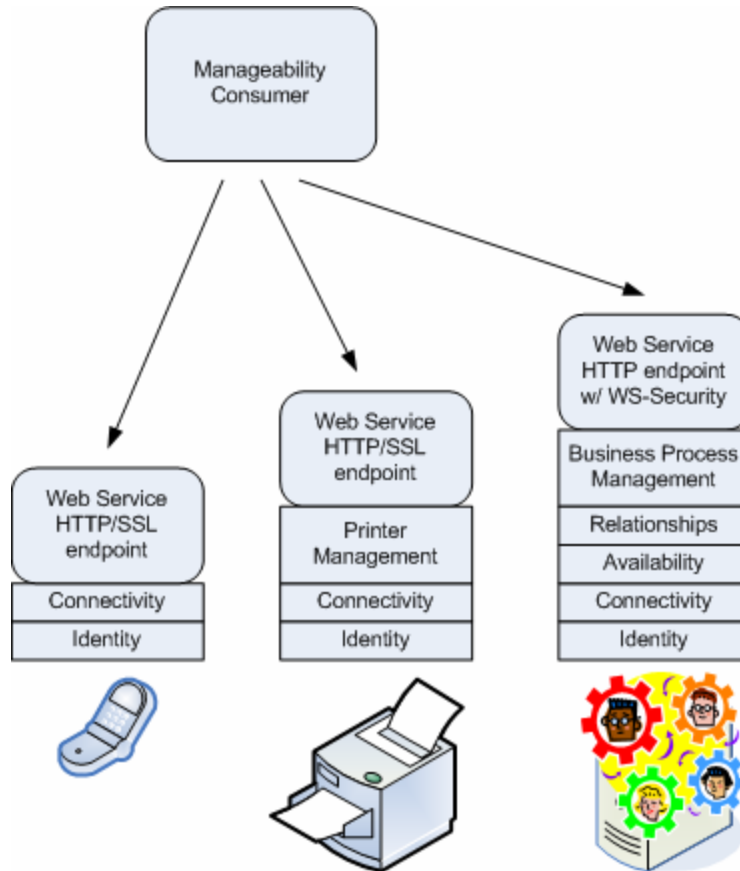
297

298

Figure 3: Composability

299 **2.2.1 Low-end to High-end Manageability**

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



310
 311 *Figure 4: Low-end to High-end Manageability*

312 **2.3 Formal Representation of the Architecture**

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

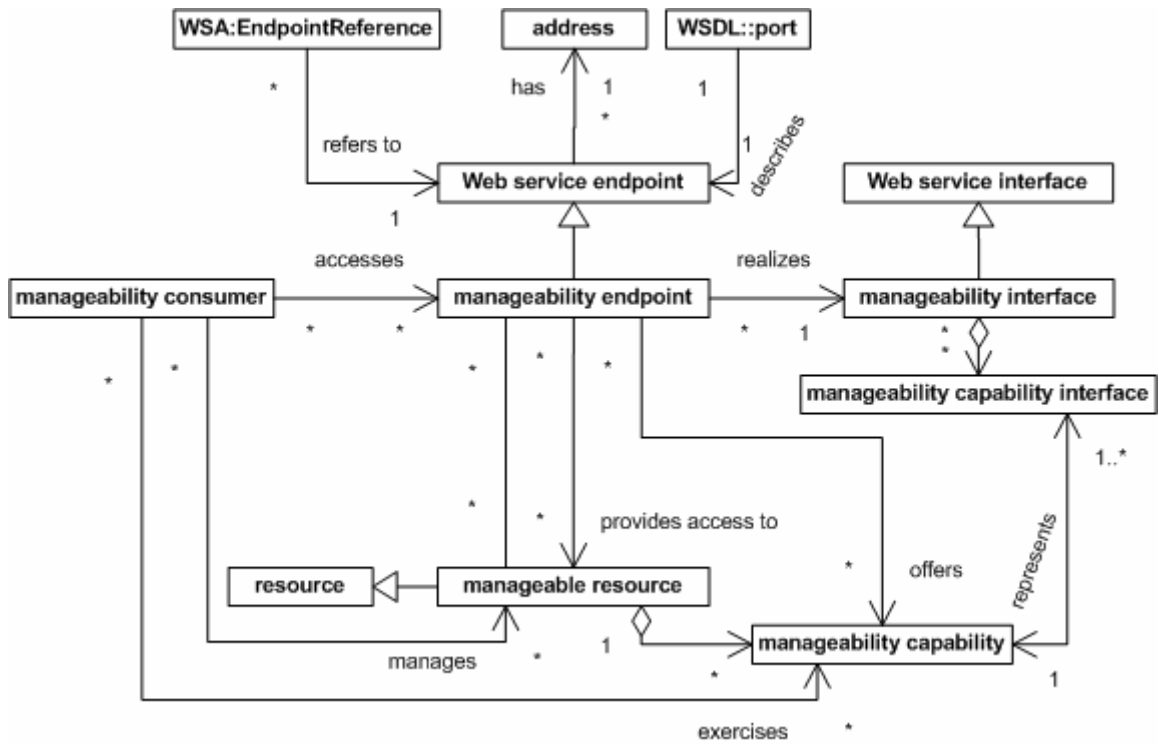


Figure 5: Formal expression of the WSDM architecture concepts

316
317
318

319 **3 Usage of the Web Services Platform**

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

326 **3.1 Properties**

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

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

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

337 **3.2 Operations**

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

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

344 **3.3 Events**

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

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

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

359 **3.4 Metadata**

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

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

370 **3.5 Addressing**

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

375 **3.6 Security**

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

- 378 • Identification: Presentation of a claimed identity
- 379 • Authentication: Verification of proof of asserted identity
- 380 • Authorization: The information and mechanisms to allow appropriate authorized requests
381 to resources and deny unauthorized requests.
- 382 • Message Integrity: The protection of messages in a message exchange from
383 unauthorized modification.
- 384 • Data Integrity: The protection of data from unauthorized modification.
- 385 • Data confidentiality
- 386 • Trust

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

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

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

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

402 This specification defines some metadata items for management. Whenever information related
403 to management metadata is being relied on, it is important to understand the environment in
404 which the metadata is being asserted. It may be needed to provide some data integrity
405 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.

408

4 Common Information Items

409

4.1 WSDM Event Format

410

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.

415

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..

419

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

420

- the identification of the resource experiencing an event, called the source,

421

- 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.

422

423

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

424

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.

430

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

431

432

4.1.1 XML Representation of the event

433

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

434

```
<muws-p1-xs:ManagementEvent ...
  muws-p1-xs:ReportTime="xs:dateTime" ?>
436
437   <muws-p1-xs:EventId>xs:anyURI</muws-p1-xs:EventId>
438
439   <muws-p1-xs:SourceComponent ...>
440     <muws-p1-xs:ResourceId>xs:anyURI</muws-p1-xs:ResourceId> ?
441     <muws-p1-xs:ComponentAddress>{any}</muws-p1-xs:ComponentAddress> *
442     {any}*
443   </muws-p1-xs:SourceComponent>
444
445   <muws-p1-xs:ReporterComponent ...>
446     <muws-p1-xs:ResourceID>xs:anyURI</muws-p1-xs:ResourceID> ?
447     <muws-p1-xs:ComponentAddress>{any}</muws-p1-xs:ComponentAddress> *
448     {any}*
449   </muws-p1-xs:ReporterComponent> ?
450   {any}*
451 </muws-p1-xs:ManagementEvent>
```

452

Where the clauses are described as follows:

453

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

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

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

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

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

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

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

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

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

494 4.2 Manageability Endpoint Reference

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

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

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

501 5 Capabilities

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

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

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

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

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

Prefix	Namespace
muws-p1-xs	http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-muws-part1-1.0.xsd
pbm	http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-pbm-1.0.xsd
xs	http://www.w3.org/2001/XMLSchema
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing

513 5.1 Identity

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

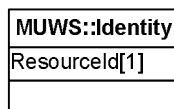
516 5.1.1 Definition

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

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

523 Figure 6 shows the UML representation of MUWS Identity.

524



525

526

Figure 6: MUWS Identity

527 5.1.2 Properties

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

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

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

wd-wsdm-muws-part1-1.0

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

Page 16 of 30


```
531 <muws-pl-xs:ResourceId>
532   http://example.com/resource/diskDrive/9F34AD35B
533 </muws-pl-xs:ResourceId>
```

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

536 This property has the following metadata:

537 It is not *Mutable*.

538 It is not *Modifiable*.

539 Its *Capability* is "http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Identity".

540 The following constraints are applicable to *ResourceId*:

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

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

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

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

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

583 5.2 Manageability Characteristics

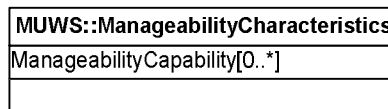
584 The manageability capability URI for the *Manageability Characteristics* capability is
585 <http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/ManageabilityCharacteristics>

586 5.2.1 Definition

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

589 Figure 7 shows the UML representation of MUWS Manageability Characteristics.

590



591

592

Figure 7: MUWS Manageability Characteristics

593 5.2.2 Properties

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

```
596 <muws-pl-xs:ManageabilityCapability>  
597   xs:anyURI  
598 </muws-pl-xs:ManageabilityCapability>*
```

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

```
601 <muws-pl-xs:ManageabilityCapability>  
602   http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/Identity  
603 </muws-pl-xs:ManageabilityCapability>  
604 <muws-pl-xs:ManageabilityCapability>  
605   http://example.com/capabilities/FooCapability  
606 </muws-pl-xs:ManageabilityCapability>
```

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

609 This property has the following metadata:

610 It is not *Mutable*.

611 It is not *Modifiable*.

612 Its *Capability* is “[http://docs.oasis-](http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/ManageabilityCharacteristics)
613 [open.org/wsdm/2004/12/muws/capabilities/ManageabilityCharacteristics](http://docs.oasis-open.org/wsdm/2004/12/muws/capabilities/ManageabilityCharacteristics)”.

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

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

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

625 5.3 Correlatable Properties

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

628 5.3.1 Definition

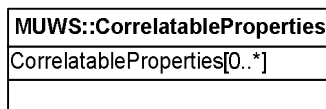
629 The *Correlatable Properties* capability allows a manageability endpoint to expose its
630 understanding of which property values could be compared when establishing that the
631 manageability endpoint in question and another manageability endpoint correspond to the same
632 resource. This is especially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

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

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

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

653



654

655

Figure 8: MUWS *Correlatable Properties*

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

662 5.3.2 Information Markup Declarations

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

wd-wsdm-muws-part1-1.0

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

Page 19 of 30

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

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

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

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

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

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

677 5.3.3 Properties

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

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

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

687 This property has the following metadata:

688 It is *Mutable*.

689 It is not *Modifiable*.

690 Its *Capability* is "http://docs.oasis-

691 open.org/wsdm/2004/12/muws/capabilities/CorrelatableProperties".

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

699 The dialects defined by this specification are:

- 700 • Simple Property Boolean Match

701 The URI for this dialect is <http://docs.oasis-open.org/wsdm/2004/12/pbm>.

702 The content of the property is as described in section 5.3.2. If all top-level match
703 conditions evaluate to true, then a correlation between manageable resources is
704 established.

- 705 • XPath 1.0

706 The URI for this dialect is <http://www.w3.org/TR/1999/REC-xpath-19991116>.

707 The content of the property is an [XPath 1.0] expression. When retrieved as a property
708 form a manageable resource, the XPath expression is evaluated on properties of another
709 manageability resource. If the XPath expression evaluates to a Boolean value of *true*, or
710 if it evaluates to a non-empty, non-boolean value, without any errors, then a correlation is
711 established between the manageable resources.

- 712 • XPath 2.0

713 The URI for this dialect is <http://www.w3.org/TR/xpath20/>.

714 The content of the property is an [XPath 2.0] expression. This XPath expression is

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

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

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

731 5.3.3.1 Examples of use

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

734 Properties obtained through manageability endpoint ME1:

```
735 <print:PrinterResourcePropDoc>  
736 ...  
737 <print:PrinterModel>PrintCo SuperJet 5000</print:PrinterModel>  
738 <print:Location>Building 42 lower pillar D4</print:Location>  
739 <print:Owner>Sir Printalot</print:Owner>  
740 <print:IPAddress>15.244.62.41</print:IPAddress>  
741 <foo:Name>Baby got ink</foo:Name>  
742 <muws-pl-xs:CorrelatableProperties  
743   Dialect="http://docs.oasis-open.org/wsdm/2004/12/pbm">  
744   <pbm:MatchAny>  
745     <pbm:Match>print:IPAddress</pbm:Match>  
746     <pbm:MatchAll>  
747       <pbm:Match>foo:Name</pbm:Match>  
748       <pbm:Match>print:PrinterModel</pbm:Match>  
749       <pbm:Match>print:Location</pbm:Match>  
750       <pbm:Match>print:Owner</pbm:Match>  
751     </pbm:MatchAll>  
752   </pbm:MatchAny>  
753 </muw-pl-xs:CorreletableProperties>  
754 </print:PrinterResourcePropDoc>
```

755 Properties obtained through manageability endpoint ME2:

```
756 <print:PrinterResourcePropDoc>  
757 ...  
758 <print:PrinterModel>PrintCo UltraJet 40</print:PrinterModel>  
759 <print:Location>Building 42 lower pillar D4</print:Location>  
760 <print:Owner>Sir Printalot</print:Owner>  
761 <print:IPAddress>15.244.10.89</print:IPAddress>  
762 <foo:Name>Baby got ink</foo:Name>  
763 </print:PrinterResourcePropDoc>
```

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

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

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

774 Properties obtained through manageability endpoint ME3:

```
775 <print:PrinterResourcePropDoc>  
776 ...  
777 <muws-p1-xs:CorrelatableProperties  
778   Dialect=http://www.w3.org/TR/1999/REC-xpath-19991116  
779   NegativeAssertionPossible="false">  
780 boolean(/print:PrinterResourcePropDoc/print:LastJob/print:JobID="5622654845  
781 1262") and  
782 boolean(/print:PrinterResourcePropDoc/print:LastJob/print:JobOriginator="15  
783 .244.30.30")  
784 </muw-p1-xs:CorrelatableProperties>  
785 </print:PrinterResourcePropDoc>
```

786 Properties obtained through manageability endpoint ME4:

```
787 <print:PrinterResourcePropDoc>  
788 ...  
789 <print:LastJob>  
790 <print:JobID>56226548451262</print:JobID>  
791 <print:JobOriginator>15.244.30.30</print:JobOriginator>  
792 <print:JobDate>2004-03-11T11:30:56Z</print:JobDate>  
793 </print:LastJob>  
794 </print:PrinterResourcePropDoc>
```

795 The *CorrelatableProperties* property, as provided through manageability endpoint ME3, asserts
796 that if a manageability endpoint provides a view of a resource for which the *JobID* of the last job is
797 56226548451262, and the *JobOriginator* of the last job is 15.244.30.30, then these manageability
798 endpoints represent the same printer. In this example, the condition is satisfied, so the consumer
799 knows that ME3 and ME4 correspond to the same physical printer. Note that, as the example
800 shows, with this dialect the consumer only needs to retrieve the *CorrelatableProperties* property
801 and no other property from ME3 to check correlation. From ME4 it needs to retrieve the
802 properties needed to evaluate the XPath expression. In this example, *NegativeAssertionPossible*
803 is set to *false*, thus a negative result would not have guaranteed that the printers behind ME3 and
804 ME4 are indeed different.

805 **6 Defining a Manageability Interface**

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

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

815 7 References

816 7.1 Normative

817 [XML1.0 3rd Edition]

818 Tim Bray, et al., *Extensible Markup Language (XML) 1.0 (Third Edition)*,
819 W3C Recommendation, February 2004, <http://www.w3.org/TR/REC-xml>

820

821 [XML Schema Part 1]

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

824

825 [XML Schema Part 2]

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

828

829 [XNS]

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

832

833 [WSDL]

833 Erik Christensen, et al., *Web services Description Language (WSDL)*
834 *1.1*, W3C Note, March 2001, <http://www.w3.org/TR/wsdl>

835

836 [WS-Addressing] Don Box, et al., *Web services Addressing (WS-Addressing)*, W3C
837 Member Submission, August 2004,

838 <http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/>

839

840 [RFC2119]

840 S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,

841 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.

842

843 [RFC2396bis]

843 T. Berners-Lee, et al., *Uniform Resource Identifier (URI): Generic*

844 *Syntax*, IETF RFC 2396bis-04, February 2004,

845 <http://www.ietf.org/internet-drafts/draft-fielding-uri-rfc2396bis-04.txt>

846

847

848

849

850

851

852

853

854

855

856

857

858

859

846 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>

860		
861	[SOAP]	Don Box, et al., <i>Simple Object Access Protocol (SOAP) 1.1</i> , W3C Note,
862		May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
863		
864	[WSA]	David Booth, et al. <i>Web Services Architecture</i> , W3C Working Group
865		Note, February 2004, http://www.w3.org/TR/2004/NOTE-ws-arch-
866		20040211/
867		
868	[WSS]	Anthony Nadalin, et al. <i>Web Services Security: SOAP Message Security</i>
869		<i>1.0</i> , OASIS Standard, March 2004, http://docs.oasis-
870		open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf
871		
872	[BP]	Keith Ballinger, et al. <i>Basic Profile Version 1.1</i> , WS-I Final Material,
873		August 2004, http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-
874		24.html

875 **Appendix A. Acknowledgements**

876 The following individuals were members of the committee when the specification was approved
877 by the technical committee

878 Guru Bhat, Jeff Bohren, Winston Bumpus, Nick Butler, Brian Carroll, Fred Carter, Michael
879 Clements, David Cox, John DeCarlo, Andreas Dharmawan, Mark Ellison, John Fuller, Paul
880 Lipton, Heather Kreger, Hal Lockhart, Frederico Maciel, Tom Maguire, Bryan Murray, Richard
881 Nikula, Mark Peel, Richard Pelavin, Homayoun Pourheidari, Warren Roberts, Karl Schopmeyer,
882 Igor Sedukhin, David Snelling, Thomas Studwell, William Vambenepe, Andrea Westerinen, Jim
883 Willits, Zhili Zhang.

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

886 **Appendix B. Notices**

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

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

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

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

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

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

Appendix C. MUWS Part 1 Schema (Normative)

916

```
917 <?xml version="1.0" encoding="utf-8"?>
918 <xs:schema
919     targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-muws-
920     part1-1.0.xsd"
921     xmlns:muws-pl-xs="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-
922     muws-part1-1.0.xsd"
923     xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
924     xmlns:xs="http://www.w3.org/2001/XMLSchema"
925     elementFormDefault="qualified" attributeFormDefault="unqualified">
926
927     <xs:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
928
929     schemaLocation="http://schemas.xmlsoap.org/ws/2004/08/addressing"/>
930
931     <xs:element name="ResourceId" type="xs:anyURI"/>
932
933     <xs:complexType name="IdentityPropertiesType">
934         <xs:sequence>
935             <xs:element ref="muws-pl-xs:ResourceId"/>
936         </xs:sequence>
937     </xs:complexType>
938
939     <xs:element name="IdentityProperties"
940         type="muws-pl-xs:IdentityPropertiesType"/>
941
942
943     <xs:element name="ManageabilityCapability" type="xs:anyURI"/>
944
945     <xs:complexType name="ManageabilityCharacteristicsPropertiesType">
946         <xs:sequence>
947             <xs:element ref="muws-pl-xs:ManageabilityCapability"
948                 minOccurs="0" maxOccurs="unbounded"/>
949         </xs:sequence>
950     </xs:complexType>
951
952     <xs:element name="ManageabilityCharacteristicsProperties"
953         type="muws-pl-xs:ManageabilityCharacteristicsPropertiesType"/>
954
955     <xs:complexType name="CorrelatablePropertiesType">
956         <xs:sequence>
957             <xs:any minOccurs="0" maxOccurs="unbounded"
958                 namespace="##other" processContents="lax"/>
959         </xs:sequence>
960         <xs:attribute name="Dialect" type="xs:anyURI"/>
961         <xs:attribute name="NegativeAssertionPossible" type="xs:boolean"/>
962         <xs:anyAttribute namespace="##other"/>
963     </xs:complexType>
964
965     <xs:element name="CorrelatableProperties"
966         type="muws-pl-xs:CorrelatablePropertiesType"/>
967
968     <xs:complexType name="CorrelatablePropertiesPropertiesType">
969         <xs:sequence>
970             <xs:element ref="muws-pl-xs:CorrelatableProperties"
971                 minOccurs="0" maxOccurs="unbounded"/>
972         </xs:sequence>
973     </xs:complexType>
974
975     <xs:element name="CorrelatablePropertiesProperties"
```

```

976         type="muws-pl-xs:CorrelatablePropertiesPropertiesType"/>
977
978
979     <xs:complexType name="ComponentAddressType">
980         <xs:sequence>
981             <xs:any namespace="##other" processContents="lax"/>
982         </xs:sequence>
983     </xs:complexType>
984
985     <xs:complexType name="ComponentType">
986         <xs:sequence>
987             <xs:element name="ResourceId" type="xs:anyURI"
988                 minOccurs="0"/>
989             <xs:element name="ComponentAddress"
990                 type="muws-pl-xs:ComponentAddressType"
991                 minOccurs="0" maxOccurs="unbounded"/>
992             <xs:any minOccurs="0" maxOccurs="unbounded"
993                 namespace="##other" processContents="lax"/>
994         </xs:sequence>
995         <xs:anyAttribute namespace="##other"/>
996     </xs:complexType>
997
998     <xs:complexType name="ManagementEventType">
999         <xs:sequence>
1000             <xs:element name="EventId" type="xs:anyURI"/>
1001             <xs:element name="SourceComponent" type="muws-pl-xs:ComponentType"/>
1002             <xs:element name="ReporterComponent" type="muws-pl-xs:ComponentType"
1003                 minOccurs="0"/>
1004             <xs:any minOccurs="0" maxOccurs="unbounded"
1005                 namespace="##other" processContents="lax"/>
1006         </xs:sequence>
1007         <xs:attribute name="ReportTime" type="xs:dateTime" use="optional"/>
1008         <xs:anyAttribute namespace="##other"/>
1009     </xs:complexType>
1010
1011     <xs:element name="ManagementEvent"
1012         type="muws-pl-xs:ManagementEventType"/>
1013
1014     <xs:element name="ManageabilityEndpointReference"
1015         type="wsa:EndpointReferenceType"/>
1016
1017 </xs:schema>
1018

```

Appendix D. Properties Boolean Match Schema (Normative)

1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
  targetNamespace="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-pbm-
1.0.xsd"
  xmlns:pbm="http://docs.oasis-open.org/wsdm/2004/12/muws/wd-wsdm-pbm-
1.0.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>
```