CIM Interop Model White Paper

CIM Version 2.7

Version 0.9 June 19, 2003

Abstract

The DMTF Common Information Model (CIM) is a conceptual information model for describing computing and business entities in enterprise and Internet environments. It provides a consistent definition and structure of data, using object-oriented techniques. The CIM Schema establishes a common conceptual framework that describes the managed environment.

The CIM Interop Model describes the common management characteristics and components of a WBEM Server. The model reflects classes and properties that are independent of any specific WBEM Server implementation.

This paper overviews the concepts that are currently modeled in CIM's Interop Schema. It is intended for WBEM Server, WBEM Client and WBEM Provider developers.

Notice

DSP0153 Status: Work-in-Progress

Table of Contents

Abstract 1
Table of Contents 3
1 Introduction 4
1.1 Overview 4
1.2 Background Reference Material 4
1.3 Terminology and Items to Note 4
1.3.1 Terms 4
1.3.2 Key Phrases 5
1.3.3 Class Names 5
2 The Interop Model 6
2.1 Overview 6
2.2 Areas Addressed by the Model 8
2.3 Object Manager Sub-Model 8
2.3.1 UML Diagram 9
2.3.2 Classes 9
2.4 Namespace Sub-Model 16
2.4.1 UML Diagram 17
2.4.2 Classes 17
2.5 Protocol Adapter Sub-Model 22
2.5.1 UML Diagram 23
2.5.2 Classes 23
2.6 WBEM Server Statistics Sub-Model 26
2.6.1 UML Diagram 26
2.6.2 Classes 26
3 Future Work 30
Appendix A – Change History 31
Appendix B – References 31

1 Introduction

1.1 Overview

The DMTF Interop Model allows WBEM implementations to be managed in an open, standard manner. The model is based on a high level logical architecture and is independent of any particular WBEM Server implementation. In other words, the concepts are applicable to ALL WBEM Server implementations. The Interop Model does not enforce any architecture requirements on implementations. It describes the infrastructure, and how WBEM Clients and WBEM Providers interact with it in an open interoperable fashion. The model provides the semantics to describe the WBEM Server infrastructure, its components and their relationships. The model allows for the management and discovery of the Server, its components, capabilities, namespaces, as well as statistics.

The high-level logical architecture and components of a WBEM Server are described, with information on how each property is to be used. This paper also describes some of the future work that the DMTF Interop Working Group is planning to address.

This paper does not describe the CIM or WBEM Specifications. It does not discuss distributed discovery, xmlCIM encoding, CIM-XML or CIM-SOAP. These topics will be described in a future WBEM InteroperabilityWhite Paper or WBEM technical notes.

1.2 Background Reference Material

The following documents can provide more information on topics that are related to the material found in this document.

Common Information Model Specification, DSP0004 / CIM Operations over HTTP, DSP0200 / Representation of CIM using XML, DSP0201 - http://www.dmtf.org/standards/published_documents.php

CIM DTD
http://www.dmtf.org/standards/standard_wbem.php

CIM Concepts White Paper, CIM Core White Paper and CIM Indications White Paper - http://www.dmtf.org/standards/published_documents.php

1.3 Terminology and Items to Note

1.3.1 Terms

For definition of WBEM terms, see Section 2.1 Overview.

1.3.2 Key Phrases

The key phrases and 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.[1]

1.3.3 Class Names

CIM class names listed in this document do not include the CIM Schema name. For example, the class CIM_ObjectManager is written as "ObjectManager". All classes are assumed to be from the CIM Schema unless otherwise noted.

[1] "Key words for Use in RFCs to Indicate Requirement Levels", RFC2119, IETF

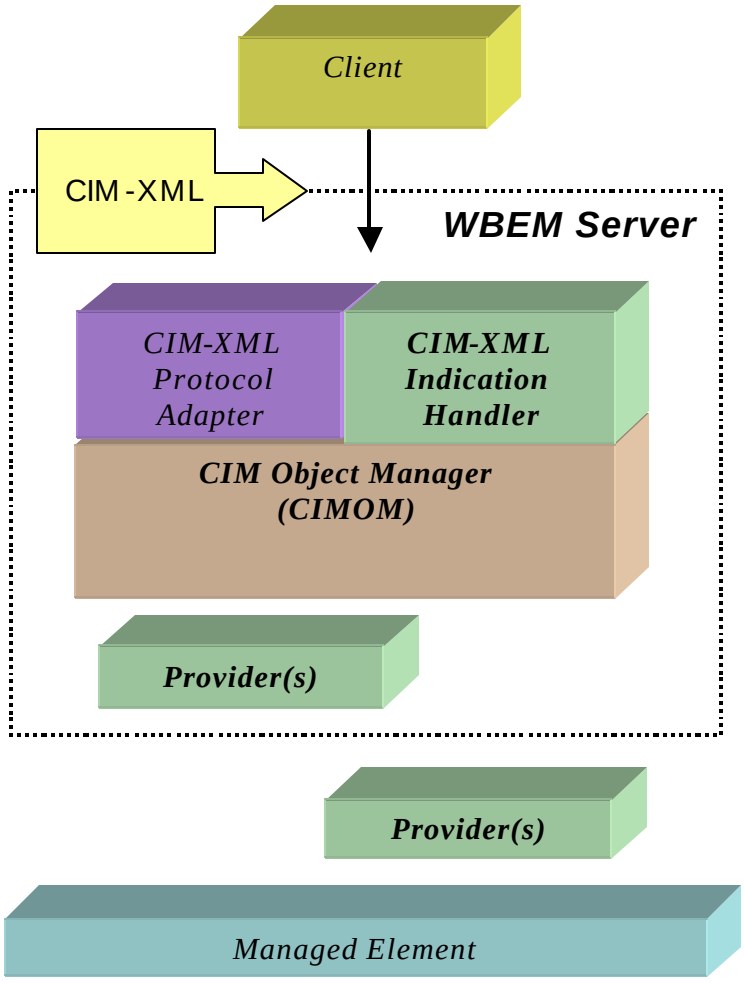
2 The Interop Model

This section describes the high level logical architecture, each Interop Sub-Model and each component in the sub-models/architecture. For each sub-model, a description is provided, with the UML diagram, the classes that make up the model and a description of each property.

2.1 Overview

The Interop Model describes the components and management aspects of a WBEM Server. The following diagram describes a high level logical architecture, and then a brief description of each component is presented.

This logical architecture does not put any architectural requirements on implementations. It is provided for the purpose of describing the functionality and components of the environment for the purposes of management and interaction with a WBEM Server. The actual implementation is left to the developer.



Term	Definition
WBEM Client	A WBEM Client issues WBEM Operation Requests and receives and processes WBEM Operation Responses. Examples of different WBEM Clients can be command line interfaces (CLI) programs, Graphical User Interface (GUI) applications, browser based applications or automated services.
CIM-XML	A WBEM Protocol defined in the CIM Operations over HTTP specification. At this time, it is the only standard WBEM protocol for exchanging CIM information.
WBEM Server	A WBEM Server is receives and processes WBEM Operation Requests and issues WBEM Operation Responses. A WBEM Server can be an agent, middle tier service or even an application. A WBEM Server MUST support basic read as defined in the CIM Operations over HTTP Specification.
WBEM Protocol Adapter	A WBEM Protocol Adapter is a component of the WBEM Server that accepts incoming requests through a particular protocol and translates these calls for the CIM Object Manager (CIMOM) and accepts responses from the CIMOM. A WBEM Protocol Adapter can be for WBEM Clients or WBEM Providers. An example of WBEM Client Protocol Adapter would be for CIM-XML.
WBEM Indication Handler	A WBEM Indication Handler is a component of the WBEM Server that delivers indications through a specified means (e.g. e-mail, pager, etc.). Indication Delivery Handlers are responsible for receiving an indication from the CIM Object Manager and delivering it to the specified destination.
CIM Object Manager	The central component of the WBEM Server responsible for the communication between all of the WBEM Server components
WBEM Provider	Anything that instruments one or more aspects of the CIM Schema.
Managed Element	Anything that needs to be managed.

2.2 Areas Addressed by the Model

The CIM Interop Model is broken down into the following models

- Object Manager Sub-Model – Describes the CIM Object Manager and its capabilities.
- Namespace Sub-Model – Describes the Namespaces available and information about what is contained in the namespace.
- Protocol Adapter Sub-Model – Describes the different protocol adapters as services and associates the applicable communication mechanisms.
- WBEM Server Statistics Sub-Model – Defines simple statistics based on WBEM operations.

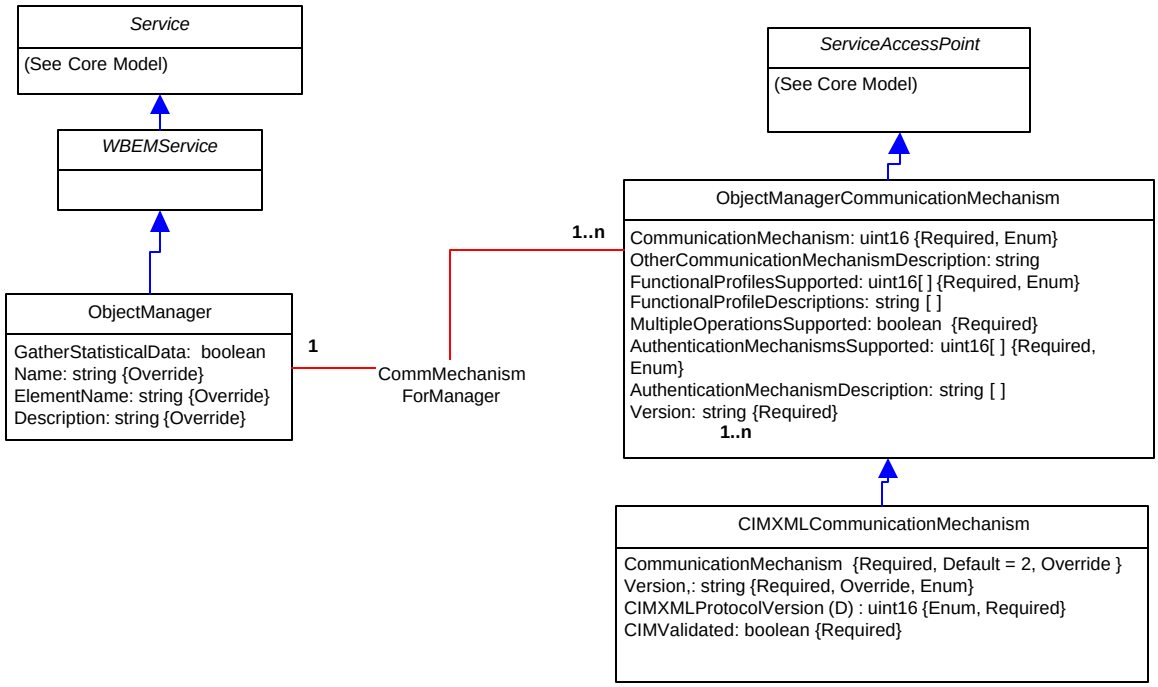
For information on areas that the DMTF Interop WG is currently working on or has plans to work on in the future see Section 3 – Future Work.

2.3 Object Manager Sub-Model

The Object Manager Sub-Model describes the WBEM Server and its capabilities. The capabilities are based on the WBEM Operations as defined in the WBEM Specifications. The capabilities are defined in the *Object Manager Communication Mechanism* class and include items such as Functional Profiles supported (e.g. Basic Read, Basic Write, …), Authentication Mechanism Supported, multiple/batched operations supported, etc.

The Object Manager Sub-Model allows for WBEM Clients to discover the communication mechanisms and capabilities of a WBEM Server. A client can enumerate the *Object Manager* class and retrieve the communication mechanism and capabilities through the C*omm Mechanism For Manager* association to the *Object Manager Communication Mechanism* class.

2.3.1 UML Diagram



2.3.2 Classes

The following table lists the classes that make up the Object Manager Sub-Model.

Class	**Superclass**
WBEMService	Service
ObjectManager	WBEMService
ObjectManagerCommunication Mechanism	ServiceAccessPoint
CIMXMLCommunicationMechanism	ObjectManagerCommunication Mechanism
CommMechanismForManager	CommMechanismForManager

2.3.2.1 WBEMService

Syntax

class CIM_WBEMService : CIM_Service

Description

The *WBEM Service* class is an abstract class for WBEM services such as the *Object Manager*, providers, CIM repositories, or any other WBEM Server component. It is a type of *Service* that provides associated capabilities and details about the component.

The *WBEM Service* class is abstract and cannot be instantiated. Since this class is abstract the properties are not listed. The properties will be listed for each concrete subclass.

2.3.2.2 ObjectManager

Syntax

class CIM_ObjectManager : CIM_WBEMService

Description

A type of service that defines the capabilities of a WBEM Server. Details related to communicating with the WBEM Server, and the Server's basic capabilities, are stored in instances of the associated *Object Mananger Communication Mechanism* class available through the *Comm Mechanism For Manager* association. WBEM Servers MUST support the Basic Read operations as defined in the CIM Operation over HTTP Specification.

Properties

Property Name	Type	Meaning
GatherStatisticalData	Boolean	The GatherStatisticalData property is used to control the gathering of statistical data made accessible through the CIM_CIMOMStatisticalData objects. If set to true, the data is gathered and can be accessed. If false, the CIM_CIMOMStatisticalData instance MAY exist but MUST show zero values for the counter properties.
SystemCreationClassName **(KEY)**	String	The scoping System's CreationClassName.
SystemName (**KEY)**	String	The scoping System's Name
CreationClassName (**KEY)**	String	CreationClassName indicates the name of the class or the subclass used in the creation of an instance.
Name **(KEY)**	String	The Name of the WBEM Server.
PrimaryOwnerName	String	The name of the primary owner for the service, if one is defined. The primary owner is the initial support contact for the Service.

Property Name	Type	Meaning
PrimaryOwnerContact	String	A string that provides information on how the primary owner of the Service can be reached (e.g. phone number, email address, ...).
StartMode	String	**(DEPRECATED)** If the WBEM Server is started automatically by the host operating or computer system, then this value MUST be Automatic, otherwise it MUST be Manual. This property MUST coincide with the EnabledDefault property for which it was deprecated.
Started	Boolean	This Value MUST be true.
EnabledState	Uint16	EnabledState is an integer enumeration that indicates the enabled/disabled states of an element. It can also indicate the transitions between these requested states. For example, shutting down (value = 4) and starting (value=8) are transient states between enabled and disabled.
OtherEnabledState	String	A string describing the element's enabled/disabled state when the EnabledState property is set to 1 ("Other").
RequestedState	Uint16	RequestedState is an integer enumeration indicating the last requested or desired state for the element.
EnabledDefault	Uint16	An enumerated value indicating an administrator's default/startup configuration for an element's EnabledState. By default, the element is "Enabled" (value=2).
InstallDate	Datetime	A datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.
OperationalStatus	Uint16[]	Indicates the current status of the element. Various health and operational statuses are defined
StatusDescriptions	String[]	Strings describing the various OperationalStatus array values.
Status	String	A string indicating the current status of the object. Various operational and non-operational statuses are defined.
Caption	String	The Caption property is a short textual description (one-line string) of the object.
Description	String	The Description property provides a textual description of the object.

Property Name	Type	Meaning
ElementName	String	A user-friendly name for the object. This property allows each instance to define a user-friendly name IN ADDITION TO its key properties/identity data, and description information.

Methods

Name	Meaning
StartService()	Since the CIM Object Manager is a special case and the WBEM Server MUST already be running, any call to this method MUST get the response CIM_ERR_METHOD_NOT_AVAILABLE
StopService()	This method MUST stop the CIM Object Manager

2.3.2.3 ObjectManagerCommunicationMechanism

Syntax
class CIM_ObjectManagerCommunicationMechanism : CIM_ServiceAccessPoint

Description

The *Object Manager Communication Mechanism* class describes access to an *Object Manager*. It describes a protocol and data encoding that can be used for communication. When all instances of this class are enumerated for an *Object Manager* (using the *Comm Mechanism For Manager* association), all possible protocol and encoding schemes will be known. Also, specific capabilities (for example, basic read, basic write, etc.) that are supported in the protocol/encoding are described - using the *Functional Profiles Supported* property.

Properties

Property Name	Type	Meaning
CommunicationMechanism **(REQUIRED)**	Uint16	CommunicationMechanism describes an encoding and protocol which can be used to communicate with the ObjectManager. At this time, only one encoding/protocol is standardized by the DMTF - 2 "CIM-XML".
OtherCommunicationMechanism	String	A free-form string providing a description of the supported encoding and protocol when 1, "Other", is specified in CommunicationMechanism.
FunctionalProfilesSupported **(REQUIRED)**	Uint16[]	Enumerated array describing the types of operations supported by the ObjectManager, using the encoding/protocol specified in the property, CommunicationMechanism.

Property Name	Type	Meaning
FunctionalProfileDescriptions	String[]	Free-form strings providing descriptions of the supported operations of the object manager. Entries in the array are correlated with those in the ProfilesSupported array. An entry in this Descriptions array MUST be provided when 1, "Other", is specified in the ProfilesSupported array.
MultipleOperationsSupported **(REQUIRED)**	Boolean	Boolean indicating whether the ObjectManager supports multiple operation requests (TRUE) or only simple requests (FALSE).
AuthenticationMechanismsSupported **(REQUIRED)**	Uint16[]	Enumerated array describing the types of authentication supported by the ObjectManager, using the encoding/protocol.
AuthenticationMechanismDescriptions	String[]	Free-form strings providing descriptions of the supported mechanisms. Entries in this array are correlated with those in the AuthenticationMechanismsSupported array. An entry in this Descriptions array MUST be provided when 1, "Other", is specified in AuthenticationMechanismsSupported.
Version **(REQUIRED)**	String	Provides the protocol version for this service access point. Version information MUST be in the form of M.N, where M is a numeric that describes the Major version and N is a numeric that describes the minor version.
CreationClassName (**KEY)**	String	CreationClassName indicates the name of the class or the subclass used in the creation of an instance.
Name **(KEY)**	String	The Name of the WBEM Server.
EnabledState	Uint16	EnabledState is an integer enumeration that indicates the enabled/disabled states of an element. It can also indicate the transitions between these requested states. For example, shutting down (value = 4) and starting (value=8) are transient states between enabled and disabled.
OtherEnabledState	String	A string describing the element's enabled/disabled state when the EnabledState property is set to 1 ("Other").
RequestedState	Uint16	RequestedState is an integer enumeration indicating the last requested or desired state for the element.

Property Name	Type	Meaning
EnabledDefault	Uint16	An enumerated value indicating an administrator's default/startup configuration for an element's EnabledState. By default, the element is "Enabled" (value=2).
InstallDate	Datetime	A datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.
OperationalStatus	Uint16[]	Indicates the current status(es) of the element. Various health and operational statuses are defined
StatusDescriptions	String[]	Strings describing the various OperationalStatus array values.
Status	String	A string indicating the current status of the object. Various operational and non-operational statuses are defined.
Caption	String	The Caption property is a short textual description (one-line string) of the object.
Description	String	The Description property provides a textual description of the object.
ElementName	String	A user-friendly name for the object. This property allows each instance to define a user-friendly name IN ADDITION TO its key properties/identity data, and description information.

Methods
There are no methods defined for this class.

2.3.2.4 CIMXMLCommunicationMechanism

Synatax

class CIM_CIMXMLCommunicationMechanism : CIM_ObjectManagerCommunicationMechanism

Description

This class specializes *Object Manager Communication Mechanism*, adding properties specific to the CIM-XML protocol (XML encoding and CIM Operations).

Properties

Note: The following table only lists the local properties (those defined or overridden in this class). To see the additional properties, refer to the superclass information (*ObjectManagerCommunicationMechanism*) in the previous section.

Property Name	Type	Meaning
CommunicationMechanism **(REQUIRED)**	Uint16	CommunicationMechanism describes an encoding and protocol which can be used to communicate with the ObjectManager. At this time, only one encoding and protocol are standardized by the DMTF - 2 "CIM-XML". If this is supported by an ObjectManager, the specified value should be indicated. In the future, other 'standard' mechanisms may be defined. In addition, a vendor specific encoding/protocol value may be specified by using the value 1, "Other", and defining the mechanism in the OtherComminicationMechanismDescription property.
Version **(REQUIRED)**	String	Provides the protocol version for this service access point. Version information MUST be in the form of M.N, where M is a numeric that describes the Major version and N is a numeric that describes the minor version.
CIMValidated **(REQUIRED)**	String	Describes whether the CIM Server is strictly validating (validates the XML document against the DTD) or not (loosely validating).
CIMXMLProtocolVersion **(DEPRECATED)**	Uint16	Enumeration describing the CIM-XML protocol version supported by the ObjectManager. It is deprecated in lieu of a more general, inherited property (Version).

Methods

There are no methods defined for this class.

2.3.2.5 CommMechanismForManager

Syntax

class CIM_CommMechanismForManager : CIM_ServiceAccessBySAP

Description

The *Comm Mechanism For Manager* is an association between an *Object Manager* and an *Object Manager Communication Mechanism* class. The latter describes a possible encoding/protocol/set of operations for accessing the referenced *Object Manager*.

Properties

Property Name	Type	Meaning
Antecedent **(KEY)**	Reference Min(1) Max(1)	The specific ObjectManager whose communication mechanism is described.
Dependent **(KEY)**	Reference Min(1)	The encoding/protocol/set of operations that may be used to communicate with the referenced ObjectManager.

Methods

There are no methods defined for this class.

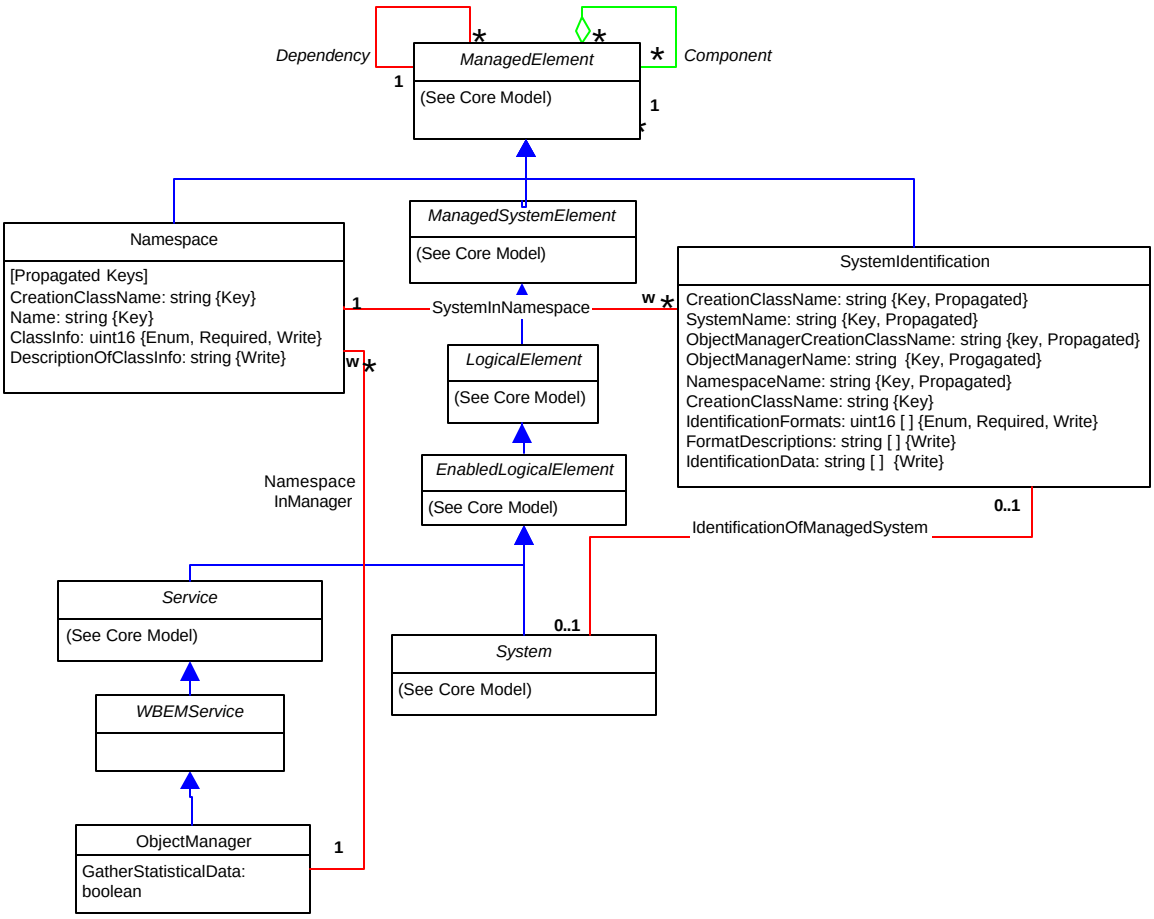
2.4 Namespace Sub-Model

The Namespace Sub-Model describes the namespaces supported by a WBEM Server as well as what type of information is contained in the namespace.

A WBEM Client can get all the namespaces supported by a WBEM Server through the *Namespace In Manager* association. Once the client has the list of supported namespaces, it can then walk the association, *System In Namespace*, to *System Identification* to determine what information is contained in the namespace.

A WBEM Client can use the Namespace class to create, delete or modify namespace information supported by a WBEM Server.

2.4.1 UML Diagram



2.4.2 Classes

The following table lists the classes that make up the Namespace Sub-Model.

Class	**Superclass**
Namespace	ManagedElement
SystemIdentification	ManagedElement
NamespaceInManager	Dependency
SystemInNamespace	
IdentificationOfManagedSystem	Dependency

2.4.2.1 Namespace

Syntax
class CIM_Namespace : CIM_ManagedElement

Description
Namespace provides a domain (in other words, a container), in which the instances [of a class] are guaranteed to be unique per the KEY qualifier definitions.

The management of namespaces is through the *Namespace* class. Using this class, you can create, delete or modify namespace information via CIM's intrinsic methods. Note that a *Namespace* instance is identified/named relative to the *Object Manager* that hosts it. A WBEM Client can find all of the namespaces supported by a WBEM Server through the *Namespace In Manager* association.

As mentioned above, Namespaces can be manipulated using CIM's intrinsic operations:

- Namespace creation – Creating a new namespace is realized by using the Create Instance operation specifying an instance of the *Namespace* class. The Name property MUST be a valid namespace name as defined in the CIM Specification.
- Namespace deletion – Deleting a namespace is realized by calling the operation Delete Instance specifying the instance of the *Namespace* desired to be removed.
- Name Modification – To modify namespace information, use the Modify Instance operation. Remember that Keys MUST NOT be modified. If the desire is to change the name of a namespace, you MUST call Delete Instance and then Create Instance. If the desire is to update (for example) Class Info (or other property that can be modified), then the Modify Instance operation MUST be used.

For more information on *Namespaces*, see the CIM Specification. For information on the rules for a WBEM Server, see the CIM Operations over HTTP Specification.

Properties

Property Name	Type	Meaning
SystemCreationClassName **(KEY)**	String	The scoping System's CreationClassName.
SystemName **(KEY)**	String	The scoping System's Name.
ObjectManagerCreationClassName **(KEY)**	String	The scoping ObjectManager's CreationClassName.
ObjectManagerName **(KEY)**	String	The scoping ObjectManager's Name.
CreationClassName **(KEY)**	String	CreationClassName indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.

Property Name	Type	Meaning
Name **(KEY)**	String	A string to uniquely identify the Namespace within the ObjectManager.
ClassInfo **(REQUIRED)**	Uint16	Enumeration indicating the organization/schema of the Namespace's objects. For example, they may be instances of classes of a specific CIM version.
DescriptionOfClassInfo	String	A string providing more detail (beyond the general classification in ClassInfo) for the object hierarchy of the Namespace.
Caption	String	The Caption property is a short textual description (one-line string) of the object.
Description	String	The Description property provides a textual description of the object.
ElementName	String	A user-friendly name for the object. This property allows each instance to define a user-friendly name IN ADDITION TO its key properties/identity data, and description information. Note that ManagedSystemElement's Name property is also defined as a user-friendly name. But, it is often subclassed to be a Key. It is not reasonable that the same property can convey both identity and a user friendly name, without inconsistencies. Where Name exists and is not a Key (such as for instances of LogicalDevice), the same information MAY be present in both the Name and ElementName properties.

2.4.2.2 SystemIdentification

Syntax
class CIM_SystemIdentification : CIM_ManagedElement

Description
A *Namespace* may represent data for one or many systems that are local, remote (different than the system on which the *Object Manager* is running) or aggregated. The *System Identification* class provides enough data to identify the system(s) represented in the *Namespace*. It is weak to the *Namespace*.

Properties

Property Name	Type	Meaning
SystemCreationClassName **(KEY)**	String	The scoping System's CreationClassName.
SystemName **(KEY)**	String	The scoping System's Name.
ObjectManagerCreationClassName **(KEY)**	String	The scoping ObjectManager's CreationClassName.
ObjectManagerName **(KEY)**	String	The scoping ObjectManager's Name.
NamespaceCreationClassName **(KEY)**	String	The scoping Namespace CreationClassName.
NamespaceName **(KEY)**	String	The scoping Namespace Name.
CreationClassName **(KEY)**	String	CreationClassName indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.
Name **(KEY)**	String	A string to uniquely identify the Namespace within the ObjectManager.
IdentificationFormats	Uint16[]	Enumeration indicating the format of the system identification and/or addressing information.
FormatDescriptions	String[]	Strings further describing the format of the system identification information.
IdentificationData	String[]	Strings containing the system identification information. The format is described by the corresponding array item in IdentificationFormats.
Caption	String	The Caption property is a short textual description (one-line string) of the object.
Description	String	The Description property provides a textual description of the object.
ElementName	String	A user-friendly name for the object. This property allows each instance to define a user-friendly name IN ADDITION TO its key properties/identity data, and description information. Note that ManagedSystemElement's Name property is also defined as a user-friendly name. But, it is often subclassed to be a Key. It is not reasonable that the same property can convey both identity and a user friendly name, without inconsistencies. Where Name exists and is not a Key (such as for instances of LogicalDevice), the same information MAY be present in both the Name and ElementName properties.

2.4.2.3 NamespaceInManager

Syntax
class CIM_NamespaceInManager : CIM_Dependency

Description
Namespace In Manager is an association describing the *Namespaces* hosted by a WBEM Server. It associates the *Object Manager* and *Namespace* classes.
AWBEM Client can determine all of the Namespaces supported by a WBEM Server by using this association to the *Object Manager*.

Properties

Property Name	Type	Meaning
Antecedent **(KEY)**	Reference	The ObjectManager containing a Namespace.
Dependent **(KEY)**	Reference	The Namespace in an ObjectManager.

2.4.2.4 SystemInNamespace

Syntax
class CIM_SystemInNamespace

Description
System In Namespace is an association that allows enumeration of the system(s) represented in a *Namespace*.

Properties

Property Name	Type	Meaning
ManagedNamespace **(KEY)**	Reference Min(1) Max(1)	The Namespace containing management objects from one or more systems.
Identification **(KEY)**	Reference	Identification information for systems in the Namespace.
ScopeOfContainedData **(REQUIRED)**	Uint16[]	A list of enumerated values providing a high level description of the data contained and allowed in the Namespace. Additional clarification is provided in the DescriptionOfContainedData array.
DescriptionOfContainedData	String[]	An array of free-form strings providing more detailed explanations for the data/objects contained in the Namespace, as described by the ContainedData array. Note, each entry of this array is related to the entry in the ContainedData array that is located at the same index.

2.4.2.5 IdentificationOfManagedSystem

Syntax
class CIM_IdentificationOfManagedSystem : CIM_Dependency

Description
Identification Of Managed System is an association that links the *System Identification* object to the CIM_System(s) that are being identified and represented in the *Namespace*.

Properties

Property Name	Type	Meaning
Antecedent	Reference Max(1)	The System which is identified.
Dependent	Reference Max(1)	The SystemIdentification information.

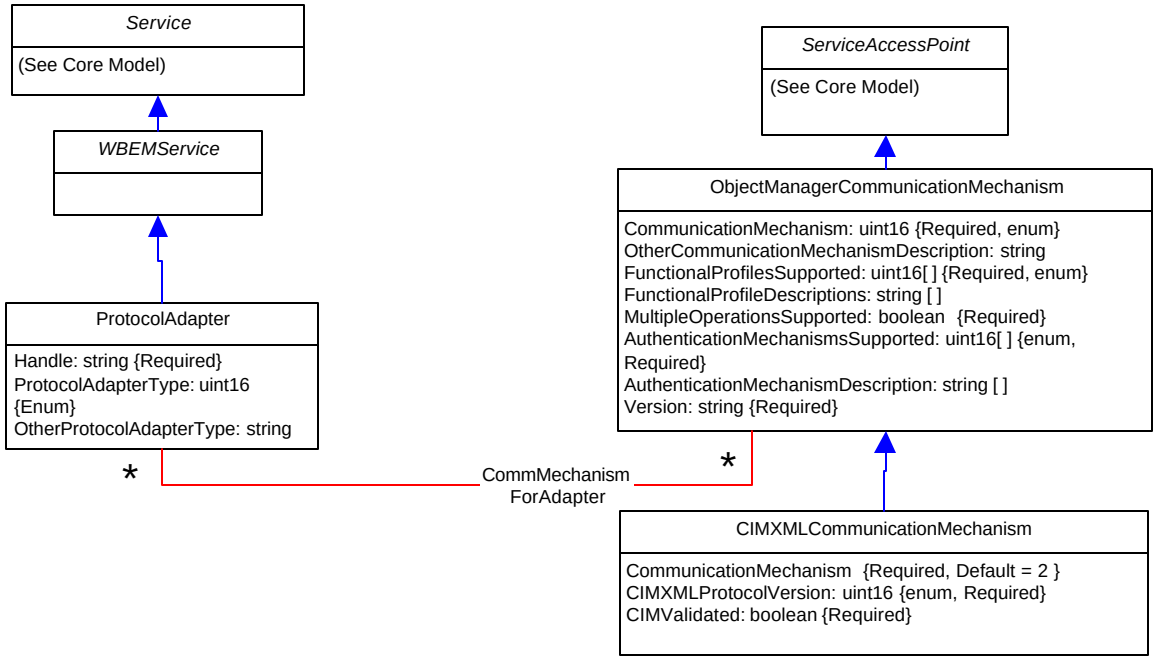
2.5 Protocol Adapter Sub-Model

The Protocol Adapter Sub-Model describes Client and Provider Protocol Adapters as services to a WBEM Server. This model is used for both registering protocol adapters with the WBEM Server, as well as administrating protocol adapters.

A WBEM Protocol Adapter SHOULD be registered through the creation of a *Protocol Adapter* instance.

A WBEM Client can find out the supported *Protocol Adapters* by using the *Comm Mechanism For Adapter* association. A WBEM Client can also start or stop the protocol adapters by using the StartService() or StopService() methods.

2.5.1 UML Diagram



2.5.2 Classes

The following table lists the classes that make up the Object Manager Model.

Class	Superclass
ProtocolAdapter	WBEMService
CommMechanismForAdapter	Dependency

2.5.2.1 Protocol Adapter

Syntax
class CIM_ProtocolAdapter : CIM_WBEMService

Description
A *Protocol Adapter* is a *Service* of the CIM *Object Manager*. It is responsible for accepting incoming requests on a particular protocol, and translating and forwarding the request to the CIM Object Manager. It is also responsible for translating and sending the response from the CIM Object Manager.

Properties

Name	Type	Meaning
SystemCreationClassName **(KEY)**	String	The scoping System's CreationClassName.
SystemName (**KEY)**	String	The scoping System's Name
CreationClassName (**KEY)**	String	CreationClassName indicates the name of the class or the subclass used in the creation of an instance.
Name **(KEY)**	String	A human-readable name that uniquely identifies the ProtocolAdapter within a system.
PrimaryOwnerName	String	The name of the primary owner for the service, if one is defined. The primary owner is the initial support contact for the Service.
PrimaryOwnerContact	String	A string that provides information on how the primary owner of the Service can be reached (e.g. phone number, email address, ...).
StartMode	String	**(DEPRECATED)** If the WBEM Server is started automatically by the host operating or computer system, then this value MUST be Automatic, otherwise it MUST be Manual. This property MUST
Started	Boolean	This Value MUST be true.
EnabledState	Uint16	EnabledState is an integer enumeration that indicates the enabled/disabled states of an element. It can also indicate the transitions between these requested states. For example, shutting down (value = 4) and starting (value=8) are transient states between enabled and disabled.
OtherEnabledState	String	A string describing the element's enabled/disabled state when the EnabledState property is set to 1 ("Other").
RequestedState	Uint16	RequestedState is an integer enumeration indicating the last requested or desired state for the element.
EnabledDefault	Uint16	An enumerated value indicating an administrator's default/startup configuration for an element's EnabledState. By default, the element is "Enabled" (value=2).
InstallDate	Datetime	A datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.
OperationalStatus	Uint16[]	Indicates the current status(es) of the element. Various health and operational statuses are defined
StatusDescriptions	String[]	Strings describing the various OperationalStatus array values.
Status	String	**(DEPRECATED)** A string indicating the current status of the object. Various operational and non-operational statuses are defined.

Name	Type	Meaning
Caption	String	The Caption property is a short textual description (one-line string) of the object.
Description	String	The Description property provides a textual description of the object.
ElementName	String	A user-friendly name for the object. This property allows each instance to define a user-friendly name IN ADDITION TO its key properties/identity data, and description information.
Handle **(REQUIRED)**	String	An implementation specific string that identifies the handle to the ProtocolAdapter. This is usually used by the WBEM Server to determine how to load/execute the protocol adapter.
ProtocolAdapterType **(REQUIRED)**	Uint16	ProtocolAdapterType enumerates the kind of ProtocolAdapter. Currently the accepted values are Other(1), Client(2) or Provider(3).
OtherProtocolAdapterType	String	The type(s) of ProtocolAdapter when "Other" is included in ProtocolAdapterType property.

Methods

Name	Meaning
StartService()	This method MUST start the Protocol Adapter
StopService()	This method MUST stop the Protocol Adapter

2.5.2.2 CommMechanismForAdapter

Syntax
Class CIM_CommMechanismForAdapter : CIM_Dependency

Description

Comm Mechanism For Adapter is an association between an *Object Manager's* communication mechanism and a *Protocol Adapter* that supports the mechanism to translate requests and responses for the Object Manager.

Properties

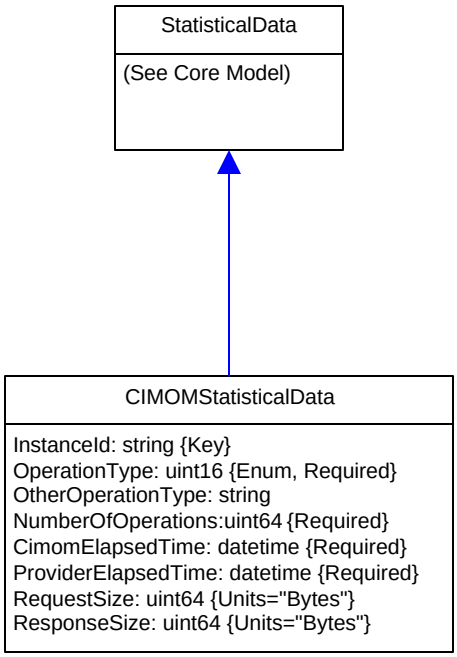
Name	Type	Meaning
Antecedent	Reference Max(1)	The specific ProtocolAdapter whose communication mechanism with the CIM Object Manager is described.
Dependent	Reference Max(1)	The encoding/protocol/set of operations that may be used to communicate between the Object Manager and the referenced ProtocolAdapter.

2.6 WBEM Server Statistics Sub-Model

The WBEM Server Statistics Sub-Model provides a set of very simple statistics for the WBEM Server, based on the WBEM Operations that have been invoked. The statistics are for the WBEM Server and can be used for diagnostic information to determine where cycles are being used.

The statistics in this model disregard the protocol or transport being used. For example, if a MOF compiler was used locally and invoked a set of Create Class operations using a local binary binding and a WBEM Client application used CIM-XML to add classes using the Create Class call, the statistics used in this model would include the combination of the all the Create Class operations.

2.6.1 UML Diagram



2.6.2 Classes

Class	Superclass
CIMOMStatisticalData	StatisticalData

2.6.2.1 CIMOMStatisticalData

Syntax
class CIM_CIMOMStatisticalData : CIM_StatisticalData

Description
CIMOM Statistical Data provides statistics about the performance of the WBEM Server. Each instance of this class provides elapsed time and size data for a particular type of CIM operation. All operations of that type, regardless of the CIM operations protocol being used, are accumulated in one object. The statistics cover the CIM operations issued by all clients of the *Object Manager* instance. Batched CIM operations are accumulated into a separate operation type for Batched.

The counters in this class SHOULD be implemented such that they always increment and naturally wrap around when their numerical limit is exceeded. A client that calculates the difference of two snapshots of a counter at the beginning and end of a measurement interval should get the correct result, even if there was a wrap-around in between obtaining the two snapshots. (Two or more wrap arounds will result in wrong data being calculated.) The gathering of the data can be controlled through the property, Object Manager. Gather Statistical Data.

The time interval to which the statistical data applies is assumed to extend to the current time. Therefore, the statistics include the most current CIM operations. The interval starts when the statistical data gathering was last turned on for the *Object Manager*.

Properties

Name	Type	Meaning
InstanceID **(KEY)**	String	The InstanceId property opaquely identifies a unique instance of CIMOMStatisticalData and MUST be unique within a namespace. In order to ensure uniqueness, the value of InstanceIdD MUST be constructed in the following manner: CIM<ID> The <ID> MUST include a CIM Object Manager specified unique identifier.
OperationType **(REQUIRED)**	Unit16	The OperationType property identifies the type of CIM operation for which data is reported in this instance. Batched CIM operations (consisting of multiple simple CIM operations) are reported against the "Batched" type only.
OtherOperationType	String	The OtherOperationType property identifies the operation if the OperationType property has a value of 1 ("Other"). For all other values of OperationType, the property is NULL.

Name	Type	Meaning
NumberOfOperations **(REQUIRED)**	Uint64	The NumberOfOperations property contains the number of CIM operations of the specified type. This property can be used to calculate average values per CIM operation.
CIMOMElapsedTime **(REQUIRED)**	Datetime	The CimomElapsedTime property contains the elapsed time spent in the CIM Object Manager for this operation type, not counting the elapsed time spent in providers and the underlying instrumentation. The measurement points for this property SHOULD be at the transport layer interface on the network side, and at the provider interface on the instrumentation side of the CIM Object Manager.
ProviderElapsedTime **(REQUIRED)**	Datetime	The ProviderElapsedTime property contains the elapsed time spent in all providers involved with this operation type, including the underlying instrumentation. The measurement point for this property SHOULD be from the provider interface of the CIM Object Manager.
RequestSize	Uint64	The RequestSize property contains the size of the operation requests sent to the CIM Object Manager. Any overhead of protocols above the transport protocol SHOULD be included in the count. For example, for the HTTP protocol, the size would include the size of the HTTP payload and the size of the HTTP headers.
ResponseSize	Uint64	The ResponseSize property contains the size of the operation responses sent back from the CIM Object Manager. Any overhead of protocols above the transport protocol SHOULD be included in the count. For example, for the HTTP protocol, the size would include the size of the HTTP payload and the size of the HTTP headers.
ElementName **(REQUIRED)**	String	The user-friendly name for this instance of StatisticalData. In addition, the user-friendly name can be used as a index property for a search of query. (Note: Name does not have to be unique within a namespace.)
Caption	String	The Caption property is a short textual description (one-line string) of the object.
Description	String	The Description property provides a textual description of the object.

Methods

Name	Meaning
ResetSelectedStats()	Method to reset one or more of the instance's statistics. The method takes one parameter as input - an array of strings indicating which statistics to reset. If all the statistics in the instance should be reset, the first element of the array MUST be set to "All" or "ALL". If one or more individual statistics should be reset, the corresponding property names are entered into the elements of the array. The method returns 0 if successful, 1 if not supported, and any other value if an error occurred. A method is specified so that the StatisticalInformation's provider/instrumentation, which calculates the statistics, can reset its internal processing, counters, etc. In a subclass, the set of possible return codes could be specified, using a ValueMap qualifier on the method. The strings to which the ValueMap contents are 'translated' may also be specified in the subclass as a Values array qualifier.

3 Future Work

The DMTF Interop WG is working on many new areas both in the areas of the model and WBEM Specifications. A list for each area is included below.

Interop Model

- Query language support (once Query spec is complete)
- Indication Handlers
- Repository
- Discovery capabilities
- Provider registration and capabilities
- Management Profiles

WBEM Specifications

- CIM-SOAP 1.0
- CIM Query Specification 1.0
- WBEM Client Operations 2.0
- CIM-XML 2.0
- Discovery (using SLP)
- WBEM Server Profile
- Representation of CIM Using XML updates
- WBEM Compliance Program

Appendix A – Change History

Version 0.9	June 19, 2003	Initial Draft

Appendix B – References

WBEMSource Open-Source Initiative, http://www.wbemsource.org/