CIM Schemas
The Core and Common Information Models

Andrea Westerinen, DMTF VP of Technology
And Work Group Chairs
Agenda

• Why CIM?
• How Schema is Defined
• Industry Efforts
• The Core Model
• The Common Models
• Development Environment
• Questions
Why CIM?

- Internet- and enterprise-wide management
  - Wide breadth of objects + repository independent
  - Unifies and extends existing standards (MIBs, X.500, M.3100, ...)
- OO design
  - Abstraction, inheritance, ability to “classify”, extensibility via subclassing
  - Well-defined “locations” and usage semantics for classes and associations
- Associations depict relationships
  - Dependencies, topologies, aggregations, scoping, ...
- “Standard”, inheritable methods
What is CIM?

• Common Information Model

• Core Specification
  – “Meta”-model, high level concepts and language definitions

• “Core” and “Common” Models
  – Object oriented design
  – Core Model contains info applicable to all management domains
  – Common Models address specific domains - Systems, Devices, Applications, Networks, Users, ...
    • Subclass from the Core Model
    • Models overlap and cross-reference
  – Vendor extensions encouraged
Definition of an Information Model

An abstraction and representation of the entities in a managed environment - their properties, operations, and relationships. It is independent of any specific repository, application, protocol, or platform.
CIM’s (OO) Approach

• Model the world in terms of objects
  – An object is an abstraction, consisting of a set of related data and behaviors
  – An object is treated by the system as a named entity that has a set of characteristics (properties and methods), behavior, and a unique identity

• Also describe relationships
  – Inheritance hierarchies refine and specialize the attributes and behavior of a group of objects
  – Association hierarchies relate objects to each other
• Associations are classes representing the relationship between two (or more) objects
  – Can have properties and methods
  – Inherit semantics, properties and behavior from superclasses
  – Can generate events/indications
• CIM distinguishes aggregation associations using a specific keyword/qualifier (“Aggregation”/”Aggregate”)
  – Represents both whole-part relationships and members-of-collections
General Example

DMTF 2002 Developers' Conference June 10-13, 2002
How is Schema Created?

CIM TC (Technical Committee)

Chair: Andrea Westerinen, Cisco
Board Members:
  Intel, Microsoft, Cisco, Sun,
  Tivoli/IBM, Compaq, Dell, HP,
  3Com, BMC, NEC, Oracle,
  Novell, Symantec
Alliance Partners, WG Chairs

Interoperability/Events
Chair: Sun

Applications/Metrics (Previously DAP)
Chair: The Open Grp

Networks
Chair: Cisco

DEN/LDAP Mapping
Chair: Cisco

PreOS
Chair: Intel

System/Devices
Chair: Cisco

Database
Chair: Oracle

Policy/SLA
Chair: IBM

User/Security
Chair: IBM

DMTF 2002 Developers’ Conference June 10-13, 2002
DMTF Development Process

- Five phases in the release of DMTF Specifications and Schema:
  - Development by Working Groups
  - Member Comment
  - Company Review
  - Preliminary Standard
  - Final Standard

- All additions and updates to CIM submitted as Change Requests to the appropriate WG and then forwarded to the Technical Committee
How Is CIM Defined?

- MOF - Managed Object Format (ASCII or Unicode)
- VISIO for UML (Unified Modeling Language)
- Whitepapers
- XML - eXtensible Markup Language
  - XML grammar can be used to describe CIM metaschema, Detailed in DTD (Document Type Defn)
    - DTD defines tags such as CLASS, INSTANCE and QUALIFIER
  - Associations are described via an ASSOCIATION.CLASS tag (Distinguished because they include references as properties)
Managed Object Format

Class Name and Inheritance

Properties

Methods

Qualifiers

[Abstract, Description (  
"An abstraction or emulation of a hardware entity, that may " 
"or may not be realized in physical hardware. ...") ]
class CIM_LogicalDevice : CIM_LogicalElement
{
    ...
    [Key, MaxLen (64), Description (  
    "An address or other identifying information to uniquely " 
    "name the LogicalDevice.") ]
    string DeviceID;
    [Description (  
    "Boolean indicating that the Device can be power " 
    "managed. ...") ]
    boolean PowerManagementSupported;
    [Description (  
    "Requests that the LogicalDevice be enabled ("Enabled") " 
    "input parameter = TRUE) or disabled (= FALSE). ...") ]
    uint32 EnableDevice([IN] boolean Enabled);
    ...
}
ASSOCIATIONS

AGGREGATION (A kind of association)

INHERITANCE

METHODS
<?XML Version="1.0"?>
<!DOCTYPE CIM SYSTEM "http://www.dmtf.org/cim-v2.dtd/">
<CIM VERSION="2.0">
<class NAME="ManagedSystemElement">
QUALIFIER NAME="abstract">
<PROPERTY NAME="Caption" TYPE="string">
<QUALIFIER NAME="MaxLen" TYPE="sint32">
<VALUE>64</VALUE></QUALIFIER>
</PROPERTY>
<PROPERTY NAME="InstallDate" TYPE="datetime">
<QUALIFIER NAME="MappingStrings" TYPE="string">
<VALUE>MIF.DMTF|ComponentID|001.5</VALUE></QUALIFIER>
</PROPERTY>
<PROPERTY NAME="Status" TYPE="string">
<QUALIFIER NAME="Values" TYPE="string" ARRAY="TRUE">
<VALUE>OK</VALUE>
<VALUE>Error</VALUE>
<VALUE>Degraded</VALUE>
<VALUE>Unknown</VALUE>
</QUALIFIER>
</PROPERTY>
<PROPERTY NAME="Description" TYPE="string">
</PROPERTY>
</CLASS>
</CIM>
Industry Adoption and Support

- **WBEMSource (open source) initiative**
  - TOG’s Pegasus and SNIA CIM Object Manager
  - Sun’s WBEM Services
  - Caldera’s OpenWBEM
  - IBM’s SBLIM and contributions to Pegasus
  - HP’s contributions to Pegasus

- **Microsoft’s WMI (Windows Mgmt Instrumentation)**

- **Sun’s WBEM Services**

- **Cisco’s CiscoWorks2000 and Policy in CNS**
Industry Adoption and Support

• SNIA (Storage Networking Industry Association)
  – Fibre Channel and Tape / Storage Library modeling and media management
  – Disk Resource Management - CIM prototype
• Coordinated development with IETF
  – Policy Framework and IPsec Policy
• The Open Group
  – Unix and software modeling, QoS and Mobile Management
  – Directory initiatives and DEN Challenge
• TM Forum
  – UML and modeling convergence
• Applications and instrumentation for the various “object managers”
Core Model

Basic Abstractions and General Concepts
Core Model – More Concepts

- ManagedElement
  - Caption : string
  - Description : string
  - ElementName : string

- Component
  - ConcreteIdentity (T)
  - ConcreteComponent (T)
  - LogicalIdentity

- ConcreteDependency (T)
- Dependency
- ConcreteDependency (T)

- Product
- FRU
- Configuration
- Setting
- SoftwareIdentity
- StatisticalData

- SupportAccess
- SettingData
- StatisticalInformation
Core – Physical Modeling and Devices

DMTF 2002 Developers’ Conference June 10-13, 2002
Core - Collections
Database Model

Goals

• Provide set of classes and associations for
  – Managing generic database systems

• Provide common classes and associations
  – Across Database organizations and vendor implementations
  – Allow vendors to extend model for their specific content

• Provide content consistent with SNMP RDBMS MIB (RFC 1697) in CIM V2.7
  – Describes common entities and properties
  – Vendor independent relational DBMS
  – Supported by many vendors

• Provide extensions for database management
  – In future CIM versions
Overview of the Database Model

• Describes a 3 Level Model for a database environment:
  – Database system software
  – Common database entity
  – Database services
    • Entities that perform tasks for database
    • Such as coordinating user access

• CIM DB model is NOT specific to RDB management
User Model

• **Related standards**
  – X.500 (Person, Group, Role, …)
  – IETF/LDAP RFCs 2252, 2256, 1274
  – NAC/Open Group’s Lightweight Internet Person

• **Model addresses:**
  – Persons and organizations
  – Groups and roles
  – Security services, credentials and accounts
  – Authentication and Access control
Group and Role Basics

- Users are members of Groups and have Roles conveying responsibilities and/or functions
- Membership is explicit via MemberOfCollection association
Organization and User Basics

**Organization**
- CreationClassName: string [Key]
- Name: string [Key] //dn
- BusinessCategory: string
- FacsimileTelephoneNumber: string
- LocalityName: string
- OU: string
- PostalAddress: string[]
- PostalCode: string
- StateOrProvince: string
- TelephoneNumber: string
- OrganizationName: string REQD

**UserEntity**
- CreationClassName: string [Key]
- Name: string [Key] //dn
- BusinessCategory: string
- EmployeeNumber: string
- EmployeeType: string
- FacsimileTelephoneNumber: string
- LocalityName: string
- Mail: string
- OrganizationName: string REQD
- PostalAddress: string[]
- PostalCode: string
- StateOrProvince: string
- TelephoneNumber: string
- ManagementControl: boolean
- PreferredLanguage: string
- Secretary: string //dn
- Title: string

**Person**
- CreationClassName: string [Key]
- Name: string [Key] //dn
- BusinessCategory: string
- EmployeeNumber: string
- EmployeeType: string
- FacsimileTelephoneNumber: string
- LocalityName: string
- Mail: string
- Manager: string //dn
- Mobile: string
- OU: string
- Pager: string
- PostalAddress: string[]
- PostalCode: string
- PreferredLanguage: string
- Secretary: string //dn
- StateOrProvince: string
- Surname: string REQD
- Title: string

**OrgUnit**
- CreationClassName: string [Key]
- Name: string [Key] //dn
- BusinessCategory: string
- FacsimileTelephoneNumber: string
- LocalityName: string
- OU: string REQD
- PostalAddress: string[]
- PostalCode: string
- StateOrProvince: string
- TelephoneNumber: string

**OrgStructure**
- CreationClassName: string [Key]
- Name: string [Key] //dn
- BusinessCategory: string
- FacsimileTelephoneNumber: string
- LocalityName: string
- OU: string
- PostalAddress: string[]
- PostalCode: string
- StateOrProvince: string
- TelephoneNumber: string

**Dependency**
- CreationClassName: string [Key]
- Name: string [Key] //dn
- BusinessCategory: string
- FacsimileTelephoneNumber: string
- LocalityName: string
- OU: string
- PostalAddress: string[]
- PostalCode: string
- StateOrProvince: string
- TelephoneNumber: string

**ElementAsUser**
- CreationClassName: string [Key]
- Name: string [Key]
- ElementID: string [Key]
- Biometric: uint16 (enum)
UsersAccess

• **Nexus of a user’s system access information**
  – For ex., credentials and system accounts independent of the associated managed element that has system access

• **Example**
  – Person may have several users accesses: administration and general business, e.g. mail
  – UsersAccess instances provide a user’s view of their relationship to systems that they use
  – ElementAsUser association conveys the ‘ownership’ relationship between the managed element that has the access and the UsersAccess instances
    • ElementID property provides name scoping
    • Name property provides a unique label for the users access instance within the scope of the ElementID
A common, declarative model

• Policies defined consistently
• Common tools & usage
• Efficient, table-driven implementations
Policy Roles provide scalability abstraction

- Resources play roles
- Policies written for roles
IndicationFilter

SystemCreationClassName: string [Key]
SystemName: string [Key]
CreationClassName: string [Key]
Name: string [Key]
SourceNamespace: string
Query: string [Required]
QueryLanguage: string [Required]

IndicationHandler

SystemCreationClassName: string [Key]
SystemName: string [Key]
CreationClassName: string [Key]
Name: string [Key]
Owner: string
PersistenceType: uint16 [enum]
OtherPersistenceType: string

IndicationSubscription

Properties provide subscription state, and error/repeat “policies”

IndicationHandlerCIMXML

Destination: string [Required]
Application Model Overview

Unit of Acquisition

Execution Service

Application Model

Collect Features into Business System

Core Model

Product

Service

Software Feature

Application System

Unit of Component Management

Unit of Deployment

ProductSoftwareFeature

Software Feature

ApplicationSystemSoftwareFeature

SoftwareFeatureSoftwareElements

Software Element
• **Checks**
  – Condition expected to be true in the environment for a Software Element
  – In-state and next-state (transition) conditions

• **Actions**
  – Operation that is part of a process to either create a Software Element in its next state or to eliminate the Software Element in the current state

• **Checks and Actions executed via the “Invoke” methods**
  – Two methods – From the original release - Invoke(), and new in V2.6 – InvokeOnSystem([IN] CIM_ComputerSystem ref TargetSystem)
Metrics Model
System Model

• Focus on ComputerSystem
• Other Systems also defined
  – StorageLibrary (CIM System)
  – AdminDomain (CIM Core)
  – ApplicationSystem (CIM Application)
• Other System Model objects:
  – OperatingSystem, Process and Thread
  – FileSystem (Local and Remote)
  – Job and JobDestination
  – MessageLogs
  – SystemResources
• New in V2.6 – UNIX model extensions
• Most classes derive from LogicalDevice
  – MediaAccessDevice and StorageExtents
  – NetworkAdapter and LogicalPort
  – Controller
  – Sensors, …
• Realize’d in hardware (Physical Elements)
• Typically have “Connection”’s to other Devices
• Associations indicate Devices’ “controlled-controller” relationships, dependencies for cooling, related software, etc.
Physical Model
Physical Model

- **Goals of the Model**
  - Inventory and asset management
  - Location of an object for service replacement or upgrade

- **Defines object hierarchy below PhysicalElement:**
  - PhysicalPackage which “contains” or hosts other components (Made explicit using the “Container” association), and may be the only PhysicalElements that have CIM_Locations
  - Subclasses of PhysicalPackage: PhysicalFrame, Rack, Chassis, Card, StorageMediaLocation
  - PhysicalLink (Cabling) and PhysicalConnector (Connection and transmission of power or signals, Slot is a subclass)
  - PhysicalComponent (Chips, PhysicalMedia, …)
  - PhysicalCapacity
Network Model

- **Focus on basic networking concepts**
  - “Cloud” and connectivity via ProtocolEndpoint and LogicalNetwork
  - Services such as SNMP, OSPF, BGP, …

- **Specific sub-models and extensions for:**
  - OSPF (Open Shortest Path First)
  - MPLS (Multi-Protocol Label Switching)
  - SNMP (Simple Network Management Protocol)
  - BGP (Border Gateway Protocol)
  - VLANs (Internal to router today, External configuration in V2.8)
  - Multiprotocol Bridges
  - QoS Services (Aligned with IETF Policy Framework WG)
  - IPSec (WIP, Aligned with IETF IPSP WG)
Other Network Concepts

• Definition of protocol stack using ProtocolEndpoints and BindsTo association
• Connections modeled via ActiveConnection association and Pipe class
• FilterLists and subclasses of FilterEntryBase (for ex., for filtering on data in the IP headers)
• RangeOfIPAddresses
• Specifics of NextHopRoute (destination and endpoint to use to transmit)
• Special admin domain - AutonomousSystem
Support Model – Solution Objects

1 Exchange Element

2 Solution

3 Categorizing the Solution

4 Adding detail and description

DMTF 2002 Developers' Conference June 10-13, 2002
Support – Admin Objects

1 Exchange Element

2 Attachments and admin info

3 Contacts and contact details

DMTF 2002 Developers' Conference June 10-13, 2002
Support – Service Incidents

1 Service

2 Incident

3 Transaction activity

4 Solutions
• Understand what data you want to manage/provide
• Where in the Schema that data fits
  – Understand the various CIM Models and locate similar concepts
  – DON’T need to take the entire model, but use the hierarchies appropriate to the problem
  – Remember that subclasses have specific semantics (such as ManagedSYSTEMElement) and inherit all properties, methods and associations from classes above
• Check/verify cardinalities on associations
• Understand “rendering” and infrastructure
  • “Rendering”
    – MOF and Visio (Today)
    – “CIM rules” (documented in the CIM Specification) – Example: It is not allowed to change data types or property names when overriding, can only change qualifiers
  • Infrastructure
    – WBEM Object Manager, LDAP directory, … influence rendering and model
    – Example: Directories defined in LDIF and do not support methods
• “CIM Core Whitepaper” and “CIM Operations over HTTP” are good starting points
Questions?