DMTF Standards Overview

Andrea Westerinen - VP of Technology, Cisco
David Lawrence – Support WG Chair, Smart Technology Enablers
Kevin Cline – ASF WG Chair, Intel
Julie Schott – SysDev WG Chair, Cisco
Topics for Discussion

• Management Environment and Architecture
• DMTF “Pieces” of the IT Puzzle
• Overview of the DMTF Standards
  – ASF, DMI and SMBIOS
  – WBEM, CIM and DEN
• Customer problems, product base and requirements are hugely variable
• Enable a collaboration of standards and vendors – Coordinate/cross-reference data and infrastructures
• More than a collection of pieces –> Architecture to enable business processes/operations and reduce operational cost
  – More than FCAPS, element managers or XML tags
• Win by having manageability/managed products that operate and interoperate in a customer’s environment
This is a view of management based on aggregating and abstracting information (higher “levels” dependent on the data from underlying ones).
• Standards and open-source are basis for infrastructure and common abstractions
  – CIM, the WBEMSource initiative, DMI, ASF, …
  – Standards ease implementation cost and enable reuse of interfaces and data
  – Open-source enables “free”, interoperable infrastructure (… there is no value-add in basic read/write capabilities)

• End results:
  – Broad model abstractions -> Coverage of multiple problem and technology domains, and simpler/more powerful mgmt products and applications
  – Abstractions and reuse of infrastructure enable new devices to be managed more rapidly, and existing app’s to manage them with minimal change
A Management Architecture

Standards (Data, Models and Protocols)

Transactions

Security

Policy

Event Correlation and Mgmt, Config, Accounting, Monitoring, and Operational Tasks

Aggregation

Interfaces – Get/Set, Create/Del, Enum ...

Info Model/Abstractions

DMTF 2002 Developers' Conference June 10-13, 2002
“Inner Circles”

- Utilize standard data and interfaces, instrumented in products
  - SMBIOS (BIOS interface on Intel-based architectures), DMI (desktop interface), ASF (pre-OS alerting), WBEM/CIM, SNMP, ...

- Enabled by a single information model (CIM)
  - Defines basic constructs (user, group, system, interface, service/functionality, …)
  - Is extensible within problem domains and for specific vendor functionality
  - Provides mechanisms to deduce semantics - For example, “know” (via inheritance) about a new FooDevice when it is defined as a subclass of LogicalDevice -> CoolingDevice
“Inner Circles”

- Interfaces defined generically such that they work at all levels
  - Manipulating data via get/set, create/delete, etc.
- Objects/instances change (or are manipulated, added onto, etc.) at the different levels
- Interfaces and model abstractions/ aggregations provide consistent semantics, as well as consistent programmatic interfaces
- Address ALL management views, tasks and levels
Mgmt Architecture from Another Angle

Integrated Mgmt Services

- System/Service Data Aggregated in CIM Repositories
- Business-wide, Directory-based Info and Policy via DEN

Managed Elements

- Desktop Mgmt Interf + Master.MIF
- WBEM + CIM
- SNMP + MIBs
- Alert Std Format - OS-Absent Envs

DMI ➔ CIM + CIM Ops + Encoding/Transport

And More

- Sys Mgmt BIOS
- DMI

DMI  ➔  CIM + CIM Ops + Encoding/Transport

Desktop Mgmt Interf + Master.MIF

WBEM + CIM

SNMP + MIBs

Alert Std Format - OS-Absent Envs
Where Do the Pieces Fit?

Management Client and Applications –
WBEM, CIM and DEN Based

DMI Technology

- DMI Management App
- DMI Service Provider
- DMI Component Instrumentation

CIM Technology

- CIM Management App
- CIM Object Manager
- CIM Provider

Sensors, Chipsets
- ASF
- SMBIOS Extensions
- Motherboard BIOS

CPU, Other Hdw
- ASF
Management Clients

• CIM unifies language and semantics of managed data; CIM object managers and providers handle access to and normalization of the managed hardware and software
  – CIM object manager can be viewed as a server, listening for requests from one or more clients and servicing those requests

• Management clients consume data and drive operations for end-to-end services
Management Clients - Possible Scenarios

- Possible client scenarios and roles:
  - Local or remote management of a single system or set of computer systems
  - Providers that act as clients – For ex., a hierarchy of CIM object managers where higher level client providers chain to lower level CIMOMs
  - Subscribers to events that may be produced by managed hardware or software anywhere in a network
  - User interfaces to computer programs that model their configuration options and functional operations as CIM Configurations or Profiles, and CIM methods
Overview of the DMTF Standards
SMBIOS

• System Management BIOS

• BIOS extension to provide system management information to local clients
  – Make, model, serial number, BIOS version, processor, memory configuration, health information and event log data

• For Intel-based architectures

• Based on work in 1994+ by BIOS vendors, Intel and Microsoft
SMBIOS Interfaces and Data

• **Table**
  – Locate data structures in upper 1 MB of real-mode memory
  – Use pointer to array of data structures
  – Every structure begins with Type, Length and Handle; Specification identifies required structures and data

• **Plug ‘n Play**
  – Locate data structure in upper 1 MB of real-mode memory
  – Use pointers in table to reach SMBIOS methods and establish protected-mode data pointer
  – Requires special access when attempted from protected mode operating systems (i.e. NT, 2K, XP)
• Desktop Management Interface
• Enables mobile, desktop and server management
• DMI Releases
  – 1.0 released in April of 1994 (Local interface only)
  – 1.1 released in January of 1996 (Fixes to 1.0)
  – 2.0 released in March of 1996 (Added remote I/F)
  – 2.0, Errata #1, released in August of 1997
  – 2.0s released in June of 1998 (Added security)
• Master.MIF updated quarterly
Simplified DMI Architecture

- **Management App**
  - MI
- **Service Provider**
  - CI
- **Instrumentation**
- **Platform specific Hardware or Software**
- **DMI (MIF) Database**
DMTF 2.0 Architecture

MA Process
Management App
Client Front End
RPC Client
Local
SP Process
SP Core
Instrumentation
CI Process
Remote Node
SP Process
RPC Server
SP Core
Instrumentation
CI Process
MIF Database
DMI Interfaces and Data

- **MI – Management Interface**
  - Provide API for management apps for local or remote access to DMI database and instrumentation

- **CI – Component Instrumentation Interface**
  - Local access only

- **Discovery**
  - `DmiRemoteRegister()` - Specified by node address (or name), RPC and transport

- **Data Tables**
  - Component – Collection of Groups
  - Group – Collection of Attributes
  - Attribute – Data value and meta data
• Alert Standard Format

• Primarily in the Ethernet controller and extends to various motherboard and system elements

• Controller collects information from system components (CPU, chipset, BIOS and sensors) and sends this to a remote server
  – System health information (for ex., POST alerts and heartbeats), environmental notifications, asset security (such as “cover tamper” and “CPU missing”)

Controller also accepts commands from the management console, and drives their execution
  – Remote power up, power down, power cycle, reset or reboot

Both “send” (alerting) and “receive” (remote control) capabilities are hardware-based and local to the networking solution

Addresses management gaps in low-power and OS-absent states (sleep, powered off, OS hung, boot)

Based on Alert on LAN (AoL) standard, developed by IBM and Intel in 1997
• Platform Event Trap (PET) to send alerts to the management console
  – SNMP Trap PDU to carry IPMI system information. The alerts cover various low-level system activities and heartbeat

• Remote Management and Control Protocol (RMCP) to do remote control of the system
  – UDP (User Datagram Protocol) based protocol used for client control functions when a managed client is in an OS-absent state
• **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
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
Customer’s Mgmt Information Stack

Applications and Services
Application Server
Database
Operating System
Systems, Devices/Storage, …
Network
Users and Security
Support
Management Infrastructure
Policy
CIM
CIM, WBEM and DEN

- “Web-Based Enterprise Management” and “Directory Enabled Networking”
- CIM to present and organize data
- Use of XML and HTTP, SOAP, LDAP, DSML and/or other web technologies

Data Description

CIM

HTTP

LDAP Directory

LDAP Access

DEN Mapping + Repository

DEN Access

WBEM Transport Encoding

WBEM Access

<xmiCIM>

DMTF 2002 Developers’ Conference June 10-13, 2002
• Web-Based Enterprise Management

• A set of technologies
  – CIM Schema
  – XML DTD to encode the Schema
  – CIM Operations over HTTP
    • HTTP/1.0 and /1.1
    • Synchronous message request and response; Simple and multiple methods supported
    • Publish/subscribe mechanism for Indications (event notifications)
    • Extrinsic (methods on a class) and intrinsic (model operations) methods are defined – Get, Create, Delete, Modify, Enumerate, …
• Directory Enabled Networks

• Map concepts from CIM (such as systems, services and policies) to a directory, and integrate this info to provide complete management architecture

• Use a directory FIRST to “direct” (register and discover) and to hold the “right” subset of management data
  – “Right” subset based on user scenarios and profiles
• Management data shared via the directory
  – DoLSuD – “DOn’t Look StUpid Data”
  – Common identity and security administration
  – Common understanding of managed systems and services
  – Locations, groupings and policy
• Agree on data based on scenarios or profiles
  – Current work is twofold - Locating management services and Support of mobility management
• CIM V1.0 submitted by Microsoft (1996), based on HyperMedia Management Schema; Revamped in CIM V2.0 (1998)
• WBEM first published in 1999 based on Interop WG specs
  – Need for an interoperable communication mechanism (to carry CIM’s interoperable data) drove the development
• DEN work based on document from the DEN Ad Hoc Working Group; Document authored by Microsoft and Cisco (1998)
  – Led to CIM Schema enhancements and the LDAP mappings
Questions?