OSI based Network Management

*Red Text* in this section highlights differences from SNMP
OSI Manager

• Responsible for management activities
  – FCAPS

• Collects and collates information received from OSI agents
OSI Agent

- Can be on a workstation or network device but must be present wherever accessing of resources is required

Duties

- Manage its own environment & interact with ‘managed objects’
- Act upon commands given by manager & accordingly manages the managed objects
- May get notifications from managed objects & forwards these to manager
Manager / Agent Relationship

Manager (Mgr) interacts with the Agent through:
- **Standard Interface** using CMIP Protocol
- **Local (proprietary) Interface**

The Agent communicates Event Reports and Management Operations & Notifications to the Agent.

Managed Objects are managed by the Agent, representing the Managed System.
Manager - Agent Interaction

• Manager & Agent exchange protocol data units
• Can be request - response interaction (used for managed operations as well as polling)
• Also unsolicited event reports (from Agent to Manager) where agent becomes aware of unusual conditions/values in managed object(s) (indicated by notifications from managed object(s) to agent)
• Also 'heartbeats', like polling except agent sends to manager at regular intervals signifying 'still alive' status of agent
CMIP Protocol Service Elements

- **M-CREATE** - directs the agent to create new instance(s) of a managed object class or attributes within a managed object
- **M-DELETE** - directs the agent to delete existing instances of managed object class(es) or attributes within a set contained in a managed object
- **M-GET** - directs the agent to return attribute values from managed objects
- **M-SET** - directs the agent to change the value of managed object attribute(s)
- **M-ACTION** - directs the agent to cause one or more managed objects to execute an action
- **M-EVENT_REPORT** - service is issued by an agent to send a notification to managers

http://www.cellsoft.de/telecom/cmip.htm
Management Information Model

- A Managed Object (MO) is a way of viewing a resource for management purposes.
- OSI MO are defined in Guidelines for Definition of Managed Objects (GDMO) language, much richer than SMI (but still based on ASN.1).
- A Managed Object is an instance of a Managed Object Class.
- Details of the managed objects are stored in a conceptual repository called a Management Information Base (MIB).
Management Information Hierarchies

OSI management Information Hierarchies

Registration Hierarchy

Containment Hierarchy

Inheritance Hierarchy
Containment Hierarchy

- One managed object can represent a part of another managed object e.g. a queue in a router.
- This is typically known as ‘containment’ and inspired the construction of global names for managed objects.
- Containment relationship is relevant to INSTANCES of managed objects.
- Containment is a directed graph where arcs connect contained and containing objects.
Object Naming

- Containment tree is used for naming managed objects
- ‘root’ is null object representing start of naming tree
- Each object instance has a relative distinguished name (RDN) which uniquely identifies the object at that level of the naming tree
- Concatenating the RDN of a object instance with those of its superiors in the naming tree will generate a globally unique object instance name
- There may be several relative distinguished names for an object => an managed object can have several globally unique names
- IMPORTANT: the containment tree and the physical containment of one resource by another need not be similar!
Scoping

- Scoping allows management operations to be performed on a selection of one or more managed objects.
- By scoping a management operation, we mean applying a management operation to a particular level or levels in the tree, starting from a specific object (base object).
- A base object can be anywhere in the naming/containment tree.
- Management operation(s) are sent to the agent in which the base object is present.
Filtering

- Filtering is an enhancement to scoping
- Used to select a subset of managed object that have been selected by scoping
  - First condition is that a specified attribute is present. If not present, the condition fails (give false as result)
  - Eight explicit matching rules: =, >=, <=, present, substring, subset of, superset of, non-null intersection
- Selection is achieved by subjecting the ‘in scope’ managed objects to condition statements yielding a BOOLEAN value
- Condition statements can be logically grouped using ‘and’, ‘or’ and ‘not’
Synchronisation

• Governs how agent performs operations on several managed objects (at one time) on behalf of a manager

• when operations such as Get, Set and Action include scoping, the issue of synchronisation arises

• Two types of synchronisation
  – Atomic: if the requested operation can be carried out on all managed objects (specified in the operation), the operation is carried out
  – Best Effort: the requested operation is carried out on as many managed objects (specified in the operation) as possible
Systems Management Operations on Objects and Attributes

Two kinds of operation on managed objects

Attribute Related Operations

- Get Attribute Value
- Replace Attribute Value
- Replace with default Value
- Add Member
- Remove Member

Managed Object Class Related Operations

- Create
- Delete
- Action
Structure of Management Information

• Managed Objects Classes are represented in an Object Oriented Hierarchy

• Each class is capable of representing attributes, operations, notifications, behaviour, inheritance relationships ....
Guidelines for definition of Managed Objects (GDMO) Templates

• ISO uses ‘templates’ for defining the components of managed object classes e.g. packages, parameters, attributes, behaviors, etc.

• Templates encourage re-use of existing definitions and therefore fewer inconsistencies between similar managed object class definitions

For more information: http://www.cellsoft.de/telecom/gdmo.htm
Templates in GDMO

- **Managed Object Class**: defines inheritance relationships, and contains packages of attributes, attribute groups, actions, & notifications.

- **Package**: Collection of attributes, attribute groups, notifications, behaviour definitions & parameter(s).

- **Attribute**: defines attribute syntax, rules to test the attribute value(s), behaviours, attribute identifier & parameters.

- **Attribute Group**: used when attributes grouped into a set for convenience. Indicates the set attributes, and an identifier value to identify the attribute group.
Templates in GDMO (cont .2)

- **Action**: defines the behavior and syntax of attribute types (which are carried in CMIS M-ACTION)

- **Behaviour**: Extends the semantics of previously defined templates in the managed object class

- **Notification**: template defines the parameters used in defining operations and notifications

- **Name binding**: template used to uniquely name a managed object. Specifies the naming attribute used for naming and identifies the superior object
Example GDMO specification of MO class

managedElement MANAGED OBJECT CLASS
DERIVED FROM "Recommendation X.721: 1992":top;
CHARACTERIZED BY
managedElementPackage PACKAGE
BEHAVIOR
managedElementBehavior BEHAVIOR
DEFINED AS
"The Managed Element object class is..."

ATTRIBUTES
managedElementId GET,
"Recommendation X.721:1992":systemTitle GET-REPLACE,
alarmStatus GET,
"Recommendation X.721:1992":administrativeState GET-REPLACE,
"Recommendation X.721:1992":operationalState GET-REPLACE,
"Recommendation X.721:1992":usageState GET-REPLACE;

NOTIFICATIONS
"Recommendation X.721:1992":environmentalAlarm,
"Recommendation X.721:1992":equipmentAlarm,
"Recommendation X.721:1992":communicationsAlarm,
"Recommendation X.721:1992":processinErrorAlarm;;

CONDITIONAL PACKAGES
createDeleteNotificationsPackage PRESENT IF "the objectCreation
and objectDeletion notifications defined in Recommendation
X.721 are supported by an instance of this class."

...;
REGISTERED AS {m3100ObjectClass 3}