

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

CONFIGURATION MANAGEMENT SERVICE DEFINITION

ECMA TR/47

July 1988

Free copies of this document are available from ECMA,
European Computer Manufacturers Association
114 Rue du Rhône – 1204 Geneva (Switzerland)

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

CONFIGURATION MANAGEMENT SERVICE DEFINITION

ECMA TR/47

July 1988

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION



TABLE OF CONTENTS

	Page
1. INTRODUCTION	1
2. SCOPE AND FIELD OF APPLICATION	1
3. REFERENCES	2
4. DEFINITIONS	5
4.1 System Administration	5
4.2 Configuration	5
4.3 OSI Resources	5
4.4 OSI-Related Resource	5
4.5 Private Resource	5
4.6 Relationship between Objects	5
4.7 Attribute	5
4.8 Object (of interest)	5
4.9 Managed Object	5
4.10 Managed Object Entry (MOE)	5
4.11 Managed Object Name (MON)	5
4.12 Managed Object Attribute (MOA)	6
4.13 Management Information Tree (MIT)	6
5. ACRONYMS	6
6. CONVENTIONS	6
7. OVERVIEW	7
7.1 Operational Background	7
7.2 Use of Common Management Information Services	9
8. CONFIGURATION MANAGEMENT MODEL	9
8.1 Operational Model of Configuration Management	9
8.2 Relationships between Objects	10
8.2.1 Types of Relationships	12
8.3 Object States and State Transitions	14
8.3.1 Object States	14
8.3.2 State Transitions	16
8.3.3 Qualification of availability	16
9. FACILITIES	19
9.1 Object Configuration Facility	19
9.2 Attribute Management Facility	19

9.3	State Management Facility	19
9.4	Relationship Management Facility	20
9.5	Software Distribution Facility	20
9.6	Object Type Registration Facility	20
10.	FUNCTIONAL UNITS	20
11.	COMMON PARAMETERS	23
11.1	InvokeId	23
11.2	ManagedObjectId	23
11.3	AccessControl	23
11.4	AttributeIdList	23
11.5	AttributeList	23
11.6	CurrentTime	23
11.7	Status	23
12.	OBJECT CONFIGURATION FACILITY	24
12.1	Create Object service	24
12.1.1	Types of Primitives and Parameters	24
12.1.2	Parameter Descriptions	24
12.2	Delete Object Service	25
12.2.1	Types of Primitives and Parameters	25
12.2.2	Parameter Descriptions	25
12.3	Rename Object Service	25
12.3.1	Types of Primitives and Parameters	25
12.3.2	Parameter Description	26
12.4	Enrolment Service	26
12.4.1	Types of Primitives and Parameters	26
12.4.2	Parameter Descriptions	26
12.5	Show Object Service	26
12.5.1	Types of Primitives and Parameters	26
12.5.2	Parameter Descriptions	27
13.	ATTRIBUTE MANAGEMENT FACILITY	27
13.1	Display Attributes service	27
13.1.1	Types of Primitives and Parameters	27
13.1.2	Parameter Description	28
13.2	Change Attributes Service	28
13.2.1	Types of Primitives and Parameters	28
13.2.2	Parameter Descriptions	28

13.3 Attributes Change Reporting Service	28
13.3.1 Types of Primitives and Parameters	28
13.3.2 Parameter Descriptions	29
14. STATE MANAGEMENT FACILITY	29
14.1 Display State service	29
14.1.1 Types of Primitives and Parameters	29
14.1.2 Parameter Descriptions	29
14.2 Change State service	30
14.2.1 Types of Primitives and Parameters	30
14.2.2 Parameter Descriptions	30
14.3 State Change Reporting service	30
14.3.1 Types of Primitives and Parameters	30
14.3.2 Parameter Descriptions	31
15. RELATIONSHIP MANAGEMENT FACILITY	31
15.1 Display Relationship Service	31
15.1.1 Types of Primitives and Parameters	31
15.1.2 Parameter Descriptions	31
15.2 Change Relationship Service	32
15.2.1 Types of Primitives and Parameters	32
15.2.2 Parameter Descriptions	32
15.3 Relationship Change Reporting service	32
15.3.1 Types of Primitives and Parameters	32
15.3.2 Parameter Descriptions	33
16. SOFTWARE DISTRIBUTION FACILITY	33
17. CONFORMANCE REQUIREMENTS	37
APPENDIX A - GENERAL DEFINITION OF THE CONCEPT OF RESOURCES AND MANAGED OBJECTS	39

1. INTRODUCTION

Management Information is used by the system administration to assist in making management decisions and to communicate those decisions to the system resources. System administration is the external user (person or machine) perspective of systems-management. It divides the management functions along logical task-related boundaries. The administrative functional areas are:

- fault management
- accounting management
- configuration and name management
- performance management
- security management

This part of this ECMA Technical Report deals with just the configuration management functional area. It addresses the requirements for configuration management from the perspective of the user of system management.

Appendix A provides for additional tutorial material on the concept of resources and managed objects and on the structure of management information (SMI). Appendix A will be part of this document until other document(s) covering SMI aspects become available.

2. SCOPE AND FIELD OF APPLICATION

The open systems management activities which are of concern are those which imply actual exchange of information between open systems. Since only the protocols needed to conduct such exchanges are candidates for standardization within the OSI architecture, this Technical Report only defines the services in terms of an abstract model. It does not imply a particular implementation of the services nor does it imply a particular representation of the service primitives.

This Technical Report is applicable to OSI management services, provided in accordance with the Basic Reference Model of Open System Interconnection. The services defined by this Technical Report are applicable to any situation, e.g. centralized or distributed, in which configuration management information is to be transferred. This Technical Report is not concerned with the interactions which result in such transfers, nor with association establishment or authorization for the transfers or for the associated activity. It is concerned solely with the form of the transfers.

The services defined by this Technical Report are based on Common Management Information Service (CMIS) as defined in ISO documents ISO/TC97/SC21/WG4 from the Tokyo meeting of June 1987. The use of other management information services if available is for further study.

This part of this Technical Report uses the descriptive conventions in the OSI Service Conventions in ISO TR8509.

3. REFERENCES

- | | |
|-------------------------|---|
| ECMA TR/37 | Framework for OSI Management |
| ISO 7498-1 | Information Processing System - Open Systems Interconnection - Basic Reference Model |
| ISO DP 7498-4 | OSI Management Framework |
| ISO TR 8509 | Information Processing Systems - Open Systems Interconnection - Service Conventions |
| ISO DP 9595-1 | Information Processing Systems - Open Systems Interconnection - Management Information Service Definition - Part 1: Overview |
| ISO DP 9595-2 | Information Processing Systems - Open Systems Interconnection - Management Information Service Definition - Part 2: Common Management Information Service |
| ISO/TC97/SC21/WG4 N 415 | Information Processing - Open System Interconnection - Management Information Service Definition - Part 5: Configuration Management Service Definition |
| ISO/TC97/SC21 N 2066 | Information Processing - Open System Interconnection - Management Information Services - Structure of Management Information: (Tokyo 1987) |

SECTION ONE

GENERAL

4. DEFINITIONS

4.1 System Administration

The external user (person or machine) perspective of systems management.

4.2 Configuration

A set of managed objects and their relationships that can be named and operated upon, as a whole, for configuration management purposes.

4.3 OSI Resources

The abstract elements which directly relate to the OSI environment as defined in the reference model.

4.4 OSI-Related Resource

The representation of OSI capabilities within a real system comprising hardware and software elements. These elements are termed OSI related resources.

4.5 Private Resource

Resources used in the real world, representing vendor or implementation specific elements.

4.6 Relationship between Objects

Relationships may exist between two or more objects. In order to perform meaningful management operations it is necessary to consider the effect that operations may have on related objects. A relationship may itself be an object.

4.7 Attribute

An Attribute is a property associated with a specific object.

4.8 Object (of interest)

Something in the world, generally the world of telecommunications and information processing, which is identifiable (can be named), and which is of interest to hold information on.

4.9 Managed Object

An object specific to the OSI environment and which has an entry in the MIB (MOE).

4.10 Managed Object Entry (MOE)

Managed Object Entry (MOE) is an entry which is the collection of information in the MIB about an instance of a managed object. It is the representation of that instance in the MIB.

4.11 Managed Object Name (MON)

A set of information that identifies a management object entry.

4.12 Managed Object Attribute (MOA)

The information of a particular type (class) concerning an object and appearing in a MOE describing the object in the MIB. An attribute has a number of component parts.

4.13 Management Information Tree (MIT)

The MIB considered as a tree, the vertices of which (other than root) are MOEs.

5. ACRONYMS

ASN.1	Abstract Syntax Notation One
ASE	Application Service Element
CMIS	Common Management Information Service
MIS	Management Information Service
CMIP	Common Management Information Protocol
SMIS	Specific Management Information Service
SMISE	Specific Management Information Service Elements
(O)SMAE	(Open) System Management Application Entity
SMP	System Management Process
MIB	Management Information Base
LM	Layer Manager

6. CONVENTIONS

This part of this Technical Report defines services for Configuration Management following the descriptive conventions defined in ISO/TR 8509.

In Section 2 of this Technical Report, the definition of each service includes a table that lists the parameters of its primitives. For a given primitive, the presence of each parameter is described by one of the following values:

M	The parameter is mandatory.
U	The use of the parameter is a service user option.
hyphen(-)	The parameter is not present in the interaction described by the primitive concerned.

7. OVERVIEW

7.1 Operational Background

An open communication facility is a collection of physical and associated logical resources that provides the ability for systems to intercommunicate. Those resources that are relevant to OSI management are called "managed objects". At any given instant in time, managed objects possess a set of properties that define and describe the managed objects at that instant. These properties comprise not only the physical nature of the managed objects and their interrelationships, but also their current state and how they are to operate.

Because the communication facility performs a set of dynamic operations, the managed objects may be in a state of constant change. Some of the properties of the managed objects can therefore change from one instant to the next, as a consequence of the operation, whether normal or abnormal, of the facility. Other properties will remain constant throughout the operation of the facility, whereas yet others will remain constant until changed by explicit management activities.

Independently of whether a property of a managed object is constant or variable, its current value may have some effect upon the current or future operation of the facility; on the other hand, it may serve only to record some aspect of the facility's current state or the history of its operation, without affecting current or future operation.

Those properties of the managed objects that are relevant to OSI operation are represented as logically structured data within the MIB and can be made accessible to system management processes. Management processes can exchange this management-related information with one another using management services. The management related information, in general

- identifies the type(class) of managed object,
- identifies the individual instance of the managed object,
- describes the relationship of managed objects to other managed objects,
- describes the characteristics of the managed object, which generally affect the operation of associated resources,
- describes the current state of the managed object, which may affect the operation of the associated resources,
- includes statistics of the past behavior of the managed object.

Configuration Management is the set of facilities which is used to assist in the management of the normal operations of an open system running in an OSI environment.

Configuration Management includes control of the initial values, reset values, and default values of managed object attributes, and management of those object attributes that govern their routine (normal) operation. Configuration Management is responsible for the setting or resetting of initial or default values for thresholds, and for the initialization and reinitialization of other attributes such as statistical counters, even though such attributes simply record previous

activity without affecting future operation (except in the sense that encroachment of a counter value upon a preset threshold may trigger a report).

The scope of configuration management facilities is:

- creation of new instances of a specific object type,
- deletion of instances of a specific object type,
- renaming of objects,
- reporting the arrival or removal of an object,
- displaying a list of object instances of a specific type,
- displaying attributes, management state, and relationships of an object,,
- changing attributes, management state, and relationships of an object,
- reporting about the change of attributes, management state, and relationships of an object,
- distribution of software.

These facilities will make use of CMIS services as defined in ISO DP 9595-2. The use of other management information transfer services is for further study.

If the information is to be communicated to another OSI system the Common Management Information Services will be used.

Configuration Management provides services that are available to other SMISE to allow them to discharge their own functional responsibilities, other SMISE are therefore users of Configuration Management. (For example, an end-to-end Fault Management facility, when replacing a failing component with a backup, can call upon the relevant Configuration Management service element to remove the failing component from service and place the backup component into service).

The demarcation line between the scope of Configuration Management and that of other SMISE is defined as follows.

Once they have been initialized or reset by Configuration Management, monitoring and other management of attributes such as statistical counters falls within the sphere of other relevant SMISE - Fault Management, Performance Management, and Accounting Management. Similarly, the setting and changing of error or load thresholds during routine operation of the system fall generally within the sphere of Fault or Performance Management, when done to fulfill their specific functions. As described above, these Service Elements may call upon the service of Configuration Management to fulfill these functions.

While Configuration Management is responsible for initial settings of configuration object attributes, it is not limited to actions that occur at system startup, since the Configuration Management model allows attributes to be changed dynamically during operation, as well as the dynamic creation of new configuration objects or the deletion of unneeded objects.

7.2 Use of Common Management Information Services

Configuration Management is a user of the CMISE (Common Management Information Service Elements). Configuration Management functions to read and set values of managed object attributes and to execute actions on managed objects (e.g. changing the object state) are performed using the M-GET, M-SET, M-CREATE, M-DELETE, and M-ACTION services, and Configuration Management reports are performed using the M-EVENT REPORT and M-CONFIRMED EVENT REPORT services of CMIS.

In some cases operations may be performed involving more than one managed object, e.g. changing the state of one managed object together with the state of the other related object or changing the relationships of related objects. The atomic control over such sets of operations, i.e. the guarantee that all changes will be performed or none at all, is the responsibility of the Configuration Management service user, using appropriate underlying services.

8. CONFIGURATION MANAGEMENT MODEL

8.1 Operational Model of Configuration Management

The model describes in an abstract way how configuration related actions on managed objects in an OSI environment will be performed by configuration management processes and how configuration related data will be communicated between open systems. The nature of OSI management is such that attributes of objects will be common to more than one of the management functional areas. For example, if an object fails, then it is quite likely that the state of that resource needs to be changed and the change information be communicated to other concerned management processes (e.g. fault management).

How the state change is achieved within one system is a local matter and not subject to standardization.

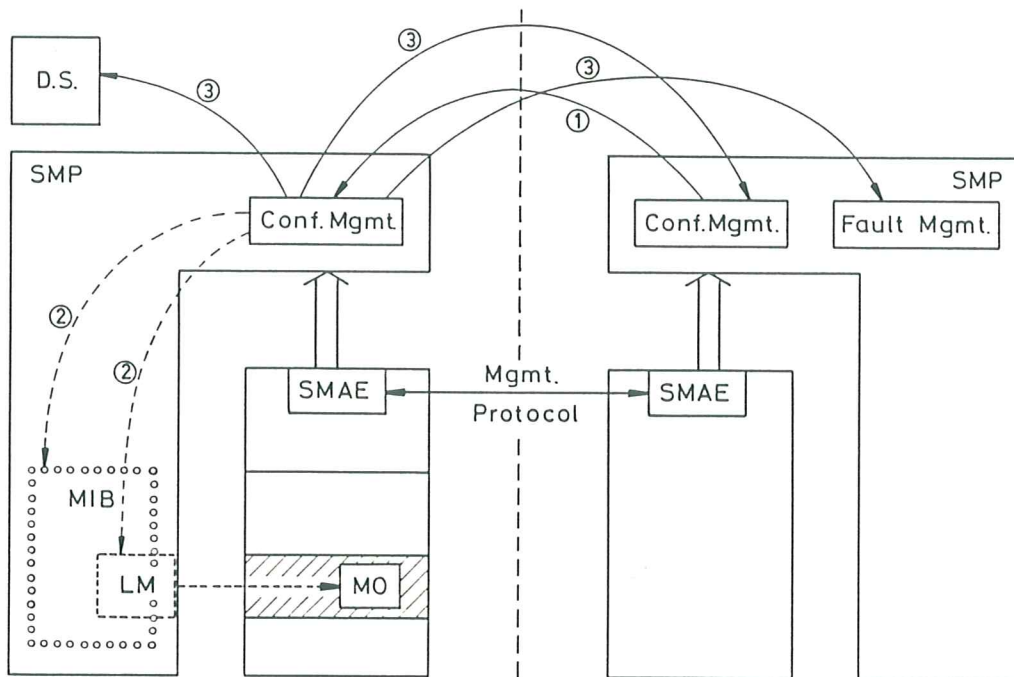


Figure 1 - Operational Model of Configuration Management Legend

Legend

- SMP System Management Process
- SMAE System Management Application Entity
- MIB Management Information Base
- LM Layer Manager
- M.O. Managed Object
- D.S. Directory Service

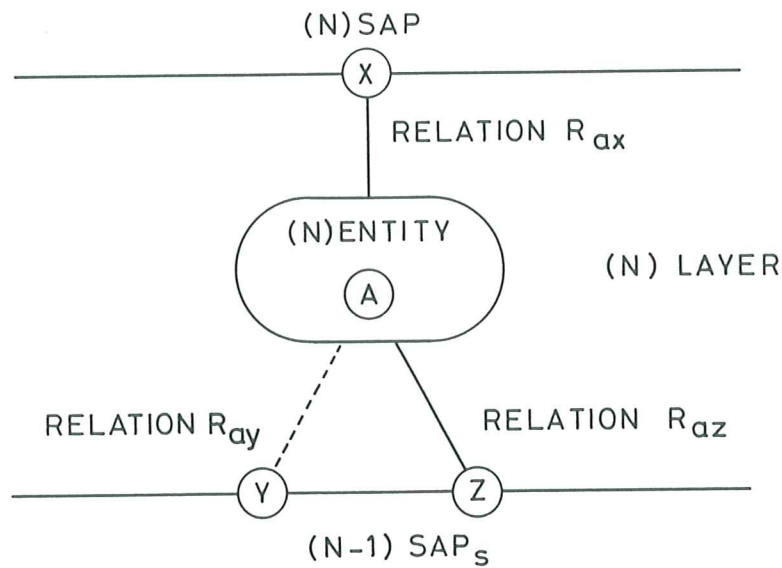
- (1) Requesting Change Operation (e.g. Change Object attribute)
- (2) Performing the change (local matter)
- (3) Reporting about the change (local and/or to another system)

8.2 Relationships between Objects

In order to perform meaningful management operations it is necessary to consider the effect that operations may have on related objects. For example the effect of, say, taking the (n)-SAP in the Figure "out of service", is to put the two related (n-1)-SAPs in the same "out of service" state, as they are inaccessible to the (n)-Service.

Therefore configuration management needs to take account of any relationships that may exist between managed objects, so that operations are performed in an orderly manner with minimum disruption to the service.

Relationships exist not only between objects associated within one OSI System, but they also may exist between objects of different OSI systems.



Jbj. Type	Obj. ID	Status	Relation
(N) SAP	X	Activated	Ray
(N) ENTITY	A	Activated	Ray, Ray, Raz
(N-1) SAP	Y	Available	Ray
(N) RELATION	Rax	Activated	A,X
(N) RELATION	Ray	Available	A,Y
(N) RELATION	Raz	Activated	A,Z
(N-1) SAP	Z	Activated	Raz

Figure 2 - Example of relationships between objects

A relationship describes an association, connection, binding or condition that exists between managed objects.

For the purpose of configuration management, relationships are regarded as management information (or attributes) that describe the association, connection, binding or condition that exists between managed objects. Additionally, relationships can be modelled as managed objects themselves.

If an object A has a relationship with another object B this relationship can be expressed in two ways:

- i) as management information describing that the A object contains a relationship attribute which points to the object B, or
- ii) as a relationship in its own right which is called a relationship object

The management information for a relationship object are:

- Name
- Status
- Attributes indicating the type of relationship (see below)
- Two or more objects between which the relationship exists.

This Technical Report provides a specific set of services for managing relationship attributes (see clause 16). However, relationship objects have to be managed by the "normal" services defined for the managing of managed objects and which are described in this Technical Report.

8.2.1 Types of Relationships

The following types of relationships are applicable to configuration management :

Ra details a relationship between a managed object and another managed object of the same type and that is capable of providing an equivalent service. The purpose of the Ra type relationship is to provide information that assists in detailing the configuration of an end system. It could be used to provide information that would assist in determining an alternative or fallback configuration in the event of a failure.

For example, two network layer X.25 protocol entities within a single end system have a relationship of a type Ra.

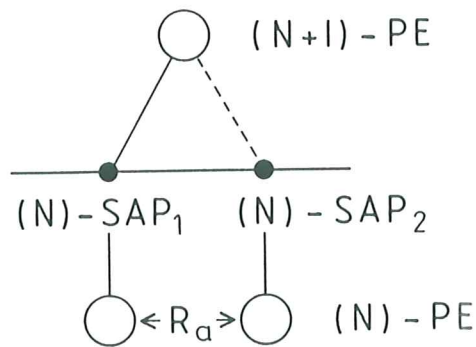


Figure 3

Rb details the "service user" to "service provider" relationships that exist between managed objects. An example of a relationship of type Rb is that of an n-SAP which has a relationship with a service user (N+1-PE) and a relationship with a service provider (N-PE).

These relationships can, as a minimum, be one-to-one but can extend to complex one-to-many relationships.

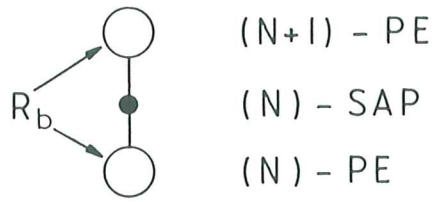


Figure 4

Rc details peer to peer relationships between managed objects residing within separate end systems.

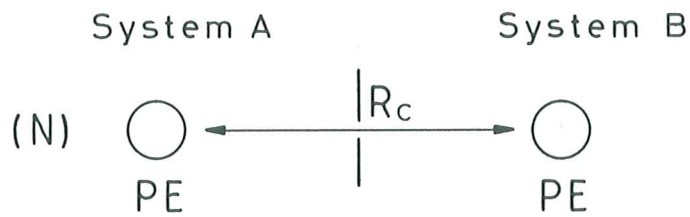


Figure 5

Rd details a topographical relationship between open systems. An example is the topography of systems within a network.

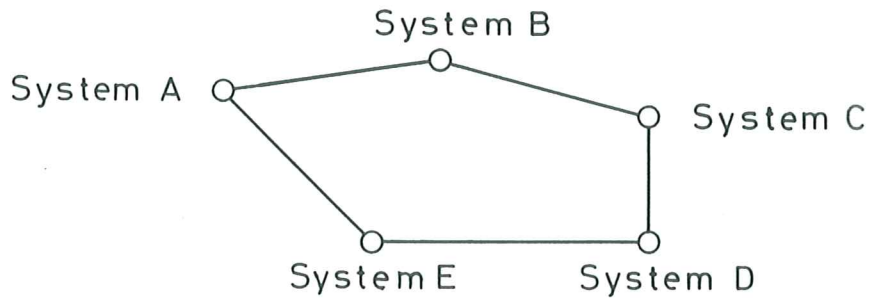


Figure 6

The following example summarises the relationships of type a) to d):

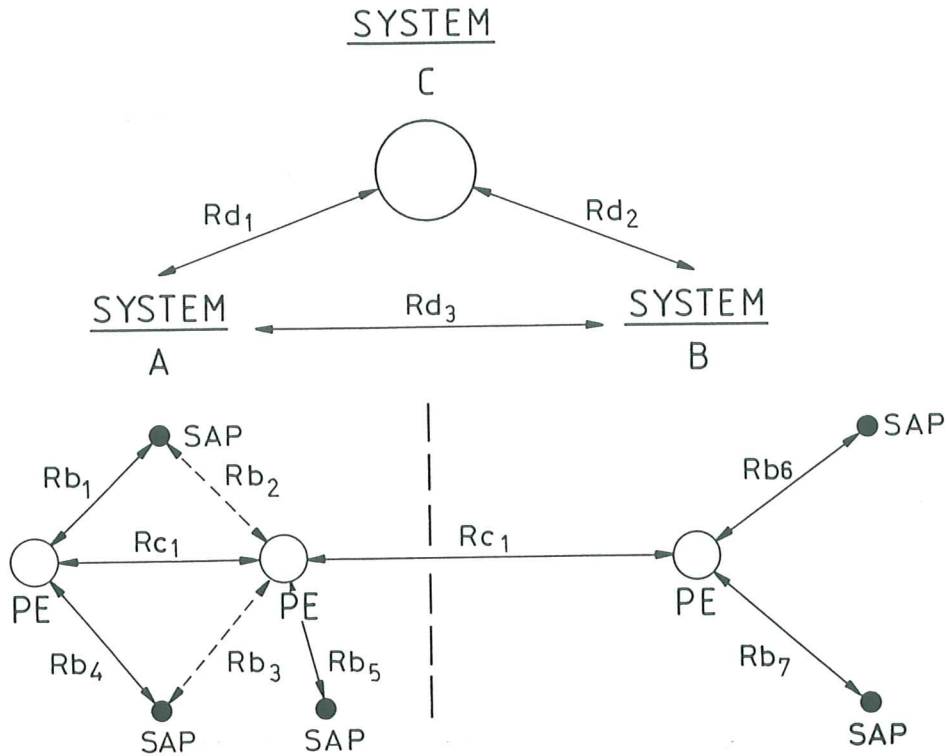


Figure 7

The relationship attribute, as described above, can consist of up to four types, each of which has a value part that details the list of managedObjectId's to which a relationship exists.

8.3 Object States and State Transitions

8.3.1 Object States

The management state of an object is an attribute representing its instantaneous condition of availability and operability from the point of view of Configuration Management Services. Different types of object have a variety of status attributes that express and control aspects of their operation that are peculiar to them. However, the management state attribute is common to all objects, and express key aspects of their usability at any given time. Its purpose is to control the general availability of an object, to make visible information about that general availability, and if an object is not usable, to indicate what action needs to be taken to make it usable.

States of managed objects fall into the following two categories:

- operational states
- administrative states.

8.3.1.1 Operational States

Operational state transitions occur as a result of the normal operation of the managed object. Management cannot directly effect control over the operational state transitions of managed objects. The operational states are

defined to enable management to monitor the operability of managed objects. Therefore operational states are read only in nature.

Operational

The managed object is operational although it is not currently being utilized.

Non-Operational

The managed object is non operational and it cannot be utilized, e.g. it is faulty or the object is not present.

Active

The managed object is administratively "in service" (see definitions for administrative control states), it is operational and it is utilized by a user or users. It has spare capacities to handle additional users.

Busy

The managed object is administratively "in service", it is operational and it is being fully utilized by a user or users. It has no spare capacity to handle additional users.

8.3.1.2 Administrative Control States

Administrative Control state transitions occur as a result of a management action. Management achieves control over the state of managed objects by manipulating the administrative control state attribute. Therefore the administrative control state attribute is read/write in nature. All managed objects that are subject to CM control may have the capability of exhibiting the following administrative control states:

In Service

The managed object is administratively regarded as available for utilization by users. Additionally, the managed object must exhibit one of the above operational states to indicate its usage.

Out of Service

The managed object is administratively regarded as being unavailable for utilization by users. Additionally, the managed object must exhibit one of the above operational states to indicate its usage.

Shutting Down

The managed object has been administratively instructed to transfer from the "In Service" to the "Out of Service" state in a "tidy" way. It will be achieved by setting the managed object in the "Shutting Down" state and continuing to process any current operations until they are completed by the user of the managed object. In the "Shutting Down" state the managed object will refuse to accept any further users. It is noted that after completion of the current operations the transition from the "Shutting Down" to the "Out of Service" state is an operational state transition.

Nonexistent

The managed object is not configured, i.e. no information about it exist in the MIB.

8.3.2 State Transitions

State transitions occur for three reasons:

- Normal operations of a system may cause an object to switch between the operational states.
- An abnormal operation of a system or an object itself may cause the object to fail and hence switch to the "Non Operational" state.
- Configuration management may initiate state transitions. Transitions out of the "Active" or "Busy" state have direct influence on the actual operation of the object. Two types of transitions are provided:
 - . The "Shutting Down" or "tidy" transition is performed in a way that has no influence on the operation. The object is marked for a transition. However, as long as the object is active or busy, the transition is not effected.
 - . The "hard" or "forced" transition is performed immediately and causes the ongoing activity to be terminated.

8.3.3 Qualification of availability

When an object is in the "available" state, it can be selected to operate. However, this availability or selectability can be qualified in the sense that the object cannot be selected in general, but is only available for a qualified usage.

Whether this qualified usage of a managed object may be considered as a subject of exerting access rights on the usage of that object or as a subject of a configuration management facility is for further study.

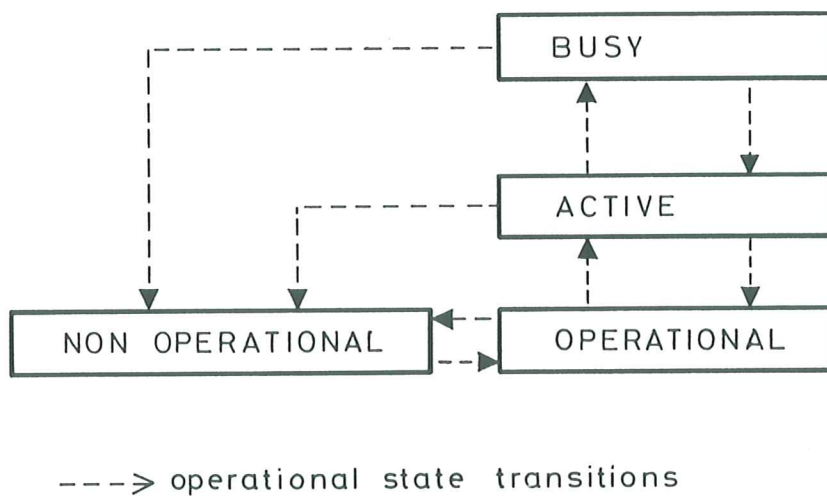
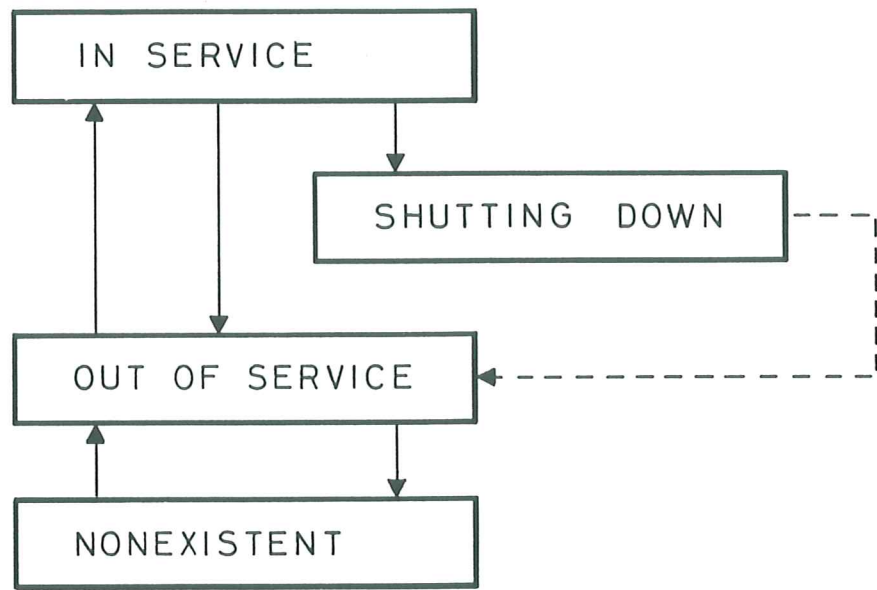


Figure 8 - Operational States



- > administrative state transitions
- - -> operational state transitions

Figure 9 - Administrative States

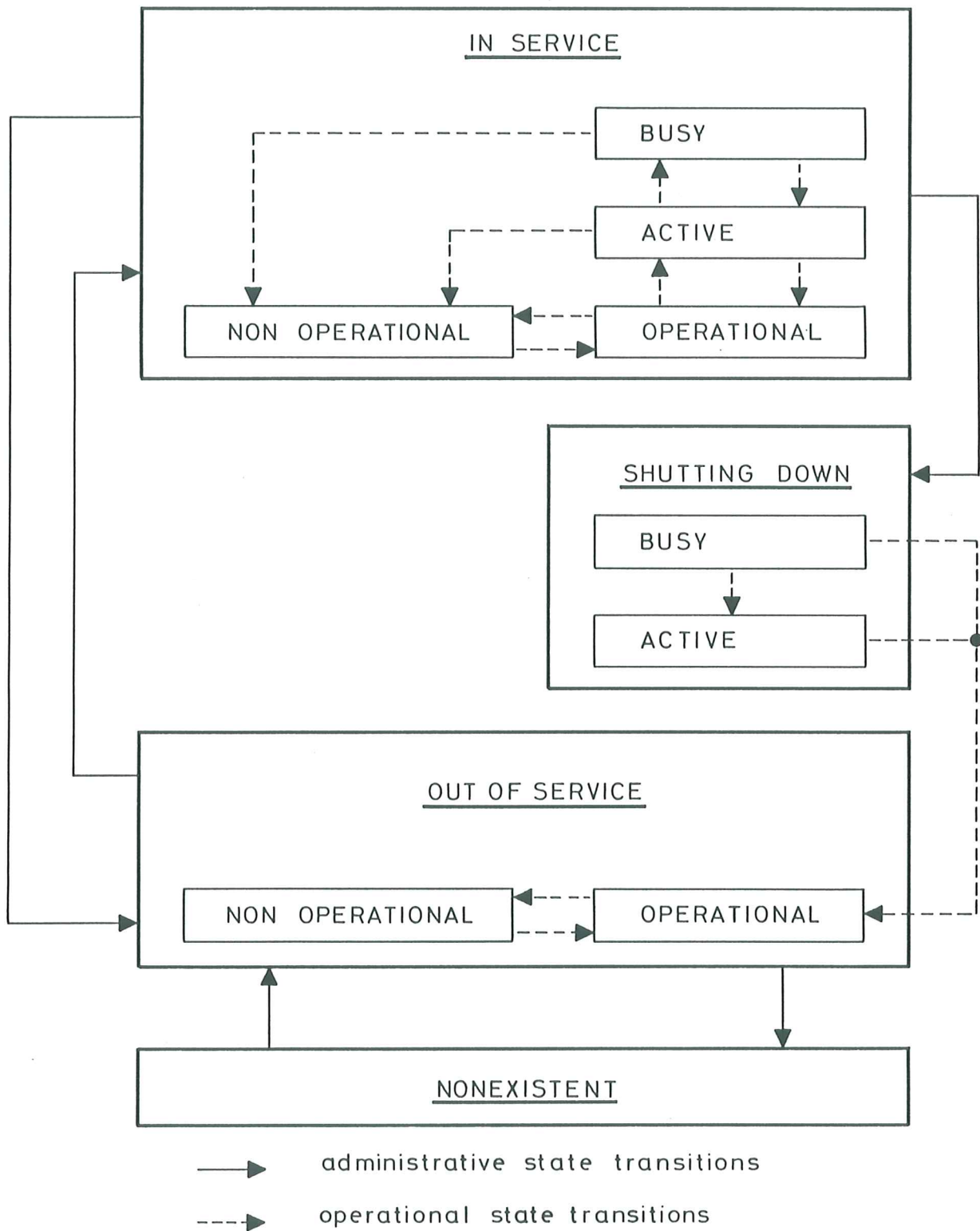


Figure 10 - Combined State Diagram

9. FACILITIES

Configuration Management provides facilities to operate upon managed objects and upon the states, relationships and attributes of managed objects. It provides facilities to obtain information, to change information and to report about changes.

9.1 Object Configuration Facility

This facility provides for the management of addition and deletion, appearance and disappearance, and renaming of object instances. It enables the invoker CM service user to request the performer CM service user that an instance of a Managed Object Entry be created from a Managed Object Type Descriptor of the appropriate type, complete with specific values as appropriate for its attributes and relationships. This facility also enables the invoker CM service user to request the performer CM service user that an instance of a Managed Object Entry be deleted from the MIB associated with the system of the performer CM service user. It enables the invoker CM service user to request the performer CM service user that an instance of a Managed Object Entry in the MIB associated with the system of the performer CM service user be renamed. It also allows the invoker CM service user to announce to the performer CM service user the presence, removal, or renaming of an object instance, and allows the invoker CM service user to request from the performer CM service user a list of object instances known at the system of the performer CM service user that comply with selection criteria provided by the invoker CM service user. This facility is provided through the CMIS M-CREATE, M-DELETE, M-GET, and M-EVENT services.

9.2 Attribute Management Facility

This facility provides for the examination, setting, and notification of changes in general attributes of existing objects. It enables the invoker CM service user to request the performer CM service user that one or more attribute values of a named object be supplied. It also enables the invoker CM service user to request the performer CM service user that one or more attributes of a named object be set to specified values. The facility also allows the invoker CM service user to announce to the performer CM service user that one or more attribute values of a named objects have changed due to some cause that may be unknown to the performer CM service user. This facility is a simple pass-through to the CMIS M-GET, M-SET, and M-EVENT-REPORT services.

9.3 State Management Facility

This facility provides for the examination, setting, and notification of changes in the management state of existing objects. It enables the invoker CM service user to request the performer CM service user that the current management state values of a named object be supplied. It also enables the invoker CM service user to request the performer CM service user that the management state values of a named object have to be changed. The facility also allows the invoker CM service user to announce to the performer CM service user that the management state values of a named object have changed due to some cause that may be unknown

to the performer CM service user. The state management facility additionally enables the invoker CM service user to request the performer CM service user that a qualified usage of an object has to be performed (reserving or releasing an object). This facility uses the CMIS M-GET, M-SET, and M-EVENT-REPORT services.

9.4 Relationship Management Facility

This facility provides for the examination, setting, and notification of changes of relationship attributes of a existing managed object. It enables the invoker CM service user to request the performer CM service user that information about the relationships of a named objects be supplied. It also enables the invoker CM service user to request the performer CM service user that the relationship attributes be set in a specified manner. The facility also allows the invoker CM service user to announce to the performer CM service user that relationship attributes have changed due to some cause that may be unknown to the performer CM service user. This facility uses the CMIS M-GET, M-SET, and M-EVENT-REPORT services.

9.5 Software Distribution Facility

For further study.

9.6 Object Type Registration Facility

This facility provides for the management of managed object type definitions.

10. FUNCTIONAL UNITS

The general service capabilities are designated as functional units, where functional units correspond to the invoker/performer support of each of the service primitives.

The functional units related to configuration management are for further study.

SECTION TWO

DEFINITION OF SERVICES

11. COMMON PARAMETERS

The following parameter definitions are common to the services defined in Section 2 of this Technical Report. These definitions are consistent with the usage of these parameters in the Common Management Information Service Definition, as specified in DP 9595-2.

11.1 InvokeId

This field specifies the identifier assigned to the operation (invocation). It can be used to distinguish the present operation from others that the invoker CM service user may have been processed by the performer CM service user.

11.2 ManagedObjectId

This field identifies the managed object to which this Configuration Management service pertains. The ManagedObjectId provides the context for the specific information in the service parameters.

11.3 AccessControl

This field is optional, it contains information of unspecified form to be used as input to access control functions in the invoker CM service user and/or the performer CM service user on any type of exchange.

11.4 AttributeIdList

This field contains a set of attributes identifiers that are to be used by the performer CM service user to read the values and return them to the invoker CM service user. The definitions for attributes are found in the layer or implementation specification selected by the ManagedObjectId of the operation.

11.5 AttributeList

This field contains a set of attribute identifiers and attribute values that are to be used by the performer CM service user to set the values of the parameters specified, or to return such values to the invoker CM service user. The definitions for attributes are found in the layer or implementation specification selected by the ManagedObjectId of the operation.

11.6 CurrentTime

For event reports, this field contains the time that the invoker CM service user generated the event report; for other services, this field contains the time that the performer CM service user generated the response.

11.7 Status

This field contains an error or other status code.

12. OBJECT CONFIGURATION FACILITY

This facility provides for the management of addition and deletion, appearance and disappearance, and renaming of object instances.

12.1 Create Object service

The Create Object service utilises the CMIS M-CREATE service to allow the invoker CM service user to request performer service user to create a managed object. The new object is created with attributes and attribute values specified within the attributeList parameter. Alternatively an object can be created with attributes and attribute values equal to another existing object using the ReferenceID parameter.

12.1.1 Types of Primitives and Parameters

Table 1 shows the types of primitives and parameters needed for creation of an object instance.

Parameter Name	Req/Ind	Resp/Conf
InvokeId	U	U
NewManagedObjectID	U	U
AttributeIdList	U	-
AttributeList	U	U
AccessControl	U	-
ReferenceID	U	-
CurrentTime	-	U
Status	-	M

The key for this table is given in clause 6 "Conventions", above.

12.1.2 Parameter Descriptions

12.1.2.1 NewmanagedObjectID

This optional parameter contains the name of the object to be created. If supplied by the invoker CM service user, it is omitted from the response. If not supplied by the invoker CM service user, a value is generated by the performer CM service user and included in the response.

12.1.2.2 AttributeIdList

This parameter contains a list of attribute identifiers related to the object to be created.

12.1.2.3 ReferenceID

This optional parameter is used by the invoker to specify that the performer is to create a new instance of a managed object, with a name specified by the parameter NewManagedObjectID, and assign attribute id and values equivalent to those associated with the managed object specified in the ReferenceID.

12.2 Delete Object Service

The Delete Object service utilises the CMIS M-DELETE service to allow the invoker CM service user to request the performer CM service user to delete an instance of a managed object. Any attributes associated with this managed object are also deleted.

12.2.1 Types of Primitives and Parameters

Table 2 shows the types of primitives and parameters needed for deletion of an object instance.

Parameter Name	Req/Ind	Resp/Conf
InvokeId	U	U
ManagedObjectID	M	U
AccessControl	U	-
CurrentTime	-	U
Status	-	M

The key for this table is given in clause 6 "Conventions", above.

12.2.2 Parameter Descriptions

The parameter definitions and sets of values given in clause 11 apply.

12.3 Rename Object Service

The Rename Object service utilized the CMIS M-Create and the CMIS M-DELETE services to allow the invoker CM service user to request the performer CM service user to rename a managed object. The CMIS M-CREATE service is used to create a duplicate of the managed object, with identical attributes and values, but with the new name. Subsequently the CMIS M-DELETE service is used to delete the instance of the managed object bearing the original name.

12.3.1 Types of Primitives and Parameters

Table 3 shows the types of primitives and parameters needed for renaming of an object instance.

Parameter Name	Req/Ind	Resp/Conf
InvokeId	U	U
ManagedObjectID	M	-
NewManagedObjectId	M	-
AccessControl	U	-
CurrentTime	-	U
Status	-	M

The key for this table is given in clause 6 "Conventions", above.

12.3.2 Parameter Description

Except as specified below, the parameter definitions and sets of values given in clause 11 apply.

12.3.2.1 NewManagedObjectId

This parameter contains the new name to be given to the object.

12.4 Enrolment Service

The Enrolment Object service utilises the CMIS M-EVENT-REPORT service to allow one CM SERVICE USER to announce the arrival (I AM ALIVE) or removal (I AM DISCONNECTING) of an object to another CM SERVICE USER.

12.4.1 Types of Primitives and Parameters

Table 4 shows the types of primitives and parameters needed for the enrolment of an object.

Parameter Name	Req/Ind	Resp/Conf
InvokeId	U	U
ManagedObjectID	M	-
CurrentTime	U	-
Status	-	M

The key for this table is given in clause 6 "Conventions", above.

12.4.2 Parameter Descriptions

The parameter definitions and sets of values given in clause 11 apply.

12.5 Show Object Service

The Show Object service utilises the CMIS M-GET service to allow the invoker CM service user to request the performer CM service user to display a list of object instances that conform to the criteria conveyed in the SelectionCriteria parameter.

12.5.1 Types of Primitives and Parameters

Table 5 shows the types of primitives and parameters needed for the show object service.

Parameter Name	Req/Ind	Resp/Conf
InvokeId	U	U
SelectionCriteria	M	-
ManagedObjectIdList	-	M
AccessControl	U	-
CurrentTime	-	U
Status	-	M

The key for this table is given in clause 6 "Conventions", above.

12.5.2 Parameter Descriptions

Except as specified below, the parameter definitions and sets of value given in clause 11 apply.

12.5.2.1 SelectionCriteria

This parameter is used by the invoker CM service user to specify criteria defining the set of managed object instances to be shown.

Note 1

The semantics of this parameter require further study. At a minimum it is necessary to be able to select a set of managed objects by object type. Additionally, the ability to select a set of objects bearing some relationship to a named object, for example, would be valuable.

12.5.2.2 ManagedObjectIdList

This parameter is used by the performer CM service user to return a list of identifiers of the managed object instances that satisfy the criteria specified in the SelectionCriteria parameter.

13. ATTRIBUTE MANAGEMENT FACILITY

This facility provides for the examination, setting and notification of changes in general attributes of existing objects.

13.1 Display Attributes service

The Display Attributes service utilises the CMIS M-GET service to allow the invoker CM service user to request the performer CM service user to return the values of one or more attributes of a managed object.

13.1.1 Types of Primitives and Parameters

Table 6 shows the types of primitives and parameters needed to display attribute values.

Parameter Name	Req/Ind	Resp/Conf
InvokeId	U	U
ManagedObjectId	M	-
AttributeIdList	M	-
AttributeList	-	M
AccessControl	U	-
CurrentTime	-	U
Status	-	M

The key for this table is given in clause 6 "Conventions", above.

13.1.2 Parameter Description

The parameter definitions and sets of values given in clause 11 apply.

13.2 Change Attributes Service

The Change Attributes Service utilises the CMIS M-SET service to allow the invoker CM service user to request the performer CM service user to change the values of the attributes of a managed object.

13.2.1 Types of Primitives and Parameters

Table 7 shows the types of primitives and parameters needed for changing the attribute values of a managed object.

Parameter Name	Req/Ind	Resp/Conf
InvokeId	U	U
ManagedObjectID	M	-
AttributeList	M	U
AccessControl	U	-
CurrentTime	-	U
Status	-	M

The key for this table is given in clause 6 "Conventions", above.

13.2.2 Parameter Descriptions

The parameter definitions and sets of values given in clause 11 apply.

13.3 Attributes Change Reporting Service

The Attributes Change Reporting Service utilises the CMIS M-EVENT-REPORT service to allow the invoker CM service user to notify the performer CM service user of changes to the values of the attributes of a managed object.

13.3.1 Types of Primitives and Parameters

Table 8 shows the types of primitives and parameters needed for reporting of changes to the values of attributes.

Parameter Name	Req/Ind	Resp/Conf
InvokeId	U	U
ManagedObjectID	M	-
AttributeId	M	-
OldValue	M	-
NewValue	M	-
CurrentTime	U	-
Status	-	M

The key for this table is given in clause 6 "Conventions", above.

13.3.2 Parameter Descriptions

Except as specified below, the parameter definitions and set of values given in clause 11 apply.

13.3.2.1 AttributeID

This parameter contains the identifier for the attribute of the managed object whose value has changed.

13.3.2.2 OldValue

This parameter contains the previous value of the attribute identified by the AttributeID parameter.

13.3.2.3 NewValue

This parameter contains the new value of the attribute identified by the AttributeID parameter.

14. STATE MANAGEMENT FACILITY

This facility provides for the examination, setting and notification of changes in the management state of existing objects.

14.1 Display State service

The Display State service utilises the CMIS M-GET service to allow the invoker CM service user to request the performer CM service user to return the values of the state attribute of a particular object.

14.1.1 Types of Primitives and Parameters

Table 9 shows the types of primitives and parameters needed to display the state of an object.

Parameter Name	Req/Ind	Resp/Conf
InvokeId	U	U
ManagedObjectID	M	-
AccessControl	U	-
CurrentState	-	M
CurrentTime	-	U
Status	-	M

The key for this table is given in clause 6 "Conventions", above.

14.1.2 Parameter Descriptions

Except as specified below, the parameter definitions and set of values given in clause 11 apply.

14.1.2.1 CurrentState

This mandatory parameter represents the state of the managed object.

14.2 Change State service

The Change State service utilises the CMIS M-SET service to allow the invoker CM service user to request the performer CM service user to change the values of the state attribute of a particular object.

14.2.1 Types of Primitives and Parameters

Table 10 shows the types of primitives and parameters needed to change the state of an object.

Parameter Name	Req/Ind	Resp/Conf
InvokeId	U	U
ManagedObjectID	M	-
NewState	M	U
AccessControl	U	-
CurrentTime	-	U
Status	-	M

The key for this table is given in clause 6 "Conventions", above.

14.2.2 Parameter Descriptions

Except as specified below, the parameter definitions and sets of value given in clause 11 apply.

14.2.2.1 NewState

This parameter is used by the invoker CM service user to specify the desired new state of the managed object. On return, it can optionally be used by the performer CM service user to specify the actual new state into which the managed object has been placed.

14.3 State Change Reporting service

The State Change Reporting service utilises the CMIS M-EVENT-REPORT service to allow the invoker CM service user to notify the CM service user of a change in the value of the state attribute of an object.

14.3.1 Types of Primitives and Parameters

Table 11 shows the types of primitives and parameters needed to notify a change in the value of the state attribute of an object.

Parameter Name	Req/Ind	Resp/Conf
InvokeId	U	U
ManagedObjectID	M	-
PreviousState	M	-
CurrentState	M	-
CurrentTime	U	-
Status	-	M

The key for this table is given in clause 6 "Conventions", above.

14.3.2 Parameter Descriptions

Except as specified below, the parameter definitions and sets of values given in clause 11 apply.

14.3.2.1 PreviousState

This mandatory parameter represents the state of the managed object prior to the transition currently being notified.

14.3.2.2 CurrentState

This mandatory parameter represents the current state of the managed object following the transition being notified.

15. RELATIONSHIP MANAGEMENT FACILITY

This facility provides for the examination, setting and notification of changes in the relationships of existing objects.

15.1 Display Relationship Service

The Display Relationship service utilises the CMIS M-GET service to allow the invoker CM service user to request the performer CM service user to return the values of the relationships attribute(s) of an object.

15.1.1 Types of Primitives and Parameters

Table 12 shows the types of primitives and parameters needed to display the relationships attribute(s) of an object.

Parameter Name	Req/Ind	Resp/Conf
InvokeId	U	U
ManagedObjectID	M	-
Relationships	-	M
AccessControl	U	-
CurrentTime	-	U
Status	-	M

The key for this table is given in clause 6 "Conventions", above.

15.1.2 Parameter Descriptions

Except as specified below, the parameter definitions and sets of values given in clause 11 apply.

15.1.2.1 Relationships

This mandatory parameter returns a list of any relationships that the specified object may have with other objects. The semantics of this parameter are for further study (see clause 8 "Model").

15.2 Change Relationship Service

The Change Relationship service utilises the CMIS M-SET service to allow the invoker CM service user to request the performer CM service user to change the values of the relationship attribute of an object.

15.2.1 Types of Primitives and Parameters

Table 13 shows the types of primitives and parameters needed for changing the relationship attribute values of a managed object.

Parameter Name	Req/Ind	Resp/Conf
InvokeId	U	U
ManagedObjectID	M	-
Relationships	M	U
AccessControl	U	-
CurrentTime	-	U
Status	-	M

The key for this table is given in clause 6 "Conventions", above.

15.2.2 Parameter Descriptions

Except as specified below, the parameter definitions and sets of values given in clause 11 apply.

1.2.2.1 Relationships

This mandatory parameter represents the relationships that the invoking CM service user requires the object to adopt. The semantics of this parameter are for further study (see clause 8 "Model").

15.3 Relationship Change Reporting service

The Relationship Change Reporting service utilises the CMIS M-EVENT-REPORT service to allow the invoker CM service user to inform the performer CM service user of a change in the value of the relationship attributes of an object.

15.3.1 Types of Primitives and Parameters

Table 14 shows the types of primitives and parameters needed for reporting of changes to the values of relationship attributes.

Parameter Name	Req/Ind	Resp/Conf
InvokeId	U	U
ManagedObjectID	M	-
PreviousRelationships	M	-
CurrentRelationships	M	-
CurrentTime	U	-
Status	-	M

The key for this table is given in clause 6 "Conventions", above.

15.3.2 Parameter Descriptions

Except as specified below, the parameter definitions and sets of values given in clause 11 apply.

15.3.2.1 PreviousRelationships

This mandatory parameter represents the relationships of the managed object prior to the change currently being notified.

15.3.2.2 CurrentRelationships

This mandatory parameter represents the relationships of the managed object following the change currently being notified.

16. SOFTWARE DISTRIBUTION FACILITY

For further study.

SECTION THREE

CONFORMANCE

17. CONFORMANCE REQUIREMENTS

This Technical Report defines a set of abstract services provided by the Configuration Management Service Elements for management of objects in open systems. As such, individual implementations and products are not specified by this Technical Report and there are no conformance requirements for this Technical Report.

APPENDIX A

GENERAL DEFINITION OF THE CONCEPT OF RESOURCES AND MANAGED OBJECTS

Configuration management activities are closely related to resources, managed objects and the attributes of these objects. This Appendix contains general definitions of resources and managed objects.

A.1 Definition of Resources

In order to structure the managed objects in a well defined manner three types of resources have been identified.

- OSI resources
- OSI related resources
- private resources

The OSI resources are those abstract elements which directly relate to the OSI environment as defined in the reference model and are listed below:

- **Open system**

An open system is itself a resource in a network of interconnected open systems

- **(N)-entities**

Within one layer, there may be different kinds of classes of entities that support different protocols

- **(N)-SAP's**

- **(N)-connections**

- **application entities**

- **relation between resources**

Each of these OSI resources are already standardized in terms of definitions. And the management information concerning these OSI resources is the subject of current standardization.

The representation of OSI capabilities within a real system (e.g. end system) comprises hardware and software elements.

These elements, which together provide an open communication facility are termed OSI related resources. Typical OSI related resources are:

- circuits
- modems and codecs
- multiplexors and switches
- supervisory systems

The management information concerning these OSI related resources is not a candidate for full standardization at the present time but it may be within the scope of standardization.

The third type of resource is termed a private resource. Private resources are only used in the real world, representing vendor or implementation specific elements. Information concerning these private resource is therefore outside the scope of standardization. However, it is meaningful to use OSI management protocols to transfer such information.

A.2 Definition of Managed Objects

A.2.1 General

Each managed object (an object specific to the OSI management) can be classified by the object type which may correspond to a specific resource or a component of it and will be identified by its individual object identification, e.g. the object name.

Additionally, an object can have one or more attributes. Attributes are a set of properties associated with a specific object. They may be simple or compound. Attributes have a type and a value.

For definition of managed objects together with their properties the following structure is suitable:

- Object Identification
- Object Type (Class)
- Attributes
 - . Information on the relationships with other objects
 - . Object state
 - . Further attributes

Object Identification

Uniquely identifies the management objects within the MIB (e.g. a specifies layer 4 protocol entity of an end system).

Object Type

Indicates the kind of object (e.g. layer 4 entity, SAP, etc.). An example of object types and their dependencies is given below.

Attributes

An object can have the following attributes:

. **Information on relationship with other objects**

expresses the relationships existing with other objects. The management of some of these relationships constitutes an important activity of Configuration Management.

. **Object state**

contains information on the status of the object.

. **Further attributes**

defines other properties of the type of object concerned

A.2.2 Identification of Managed Objects

For identification of managed objects the concepts defined in the ISO/CCITT Directory Service documents are considered. The entries referencing managed objects are depicted in the following figure:

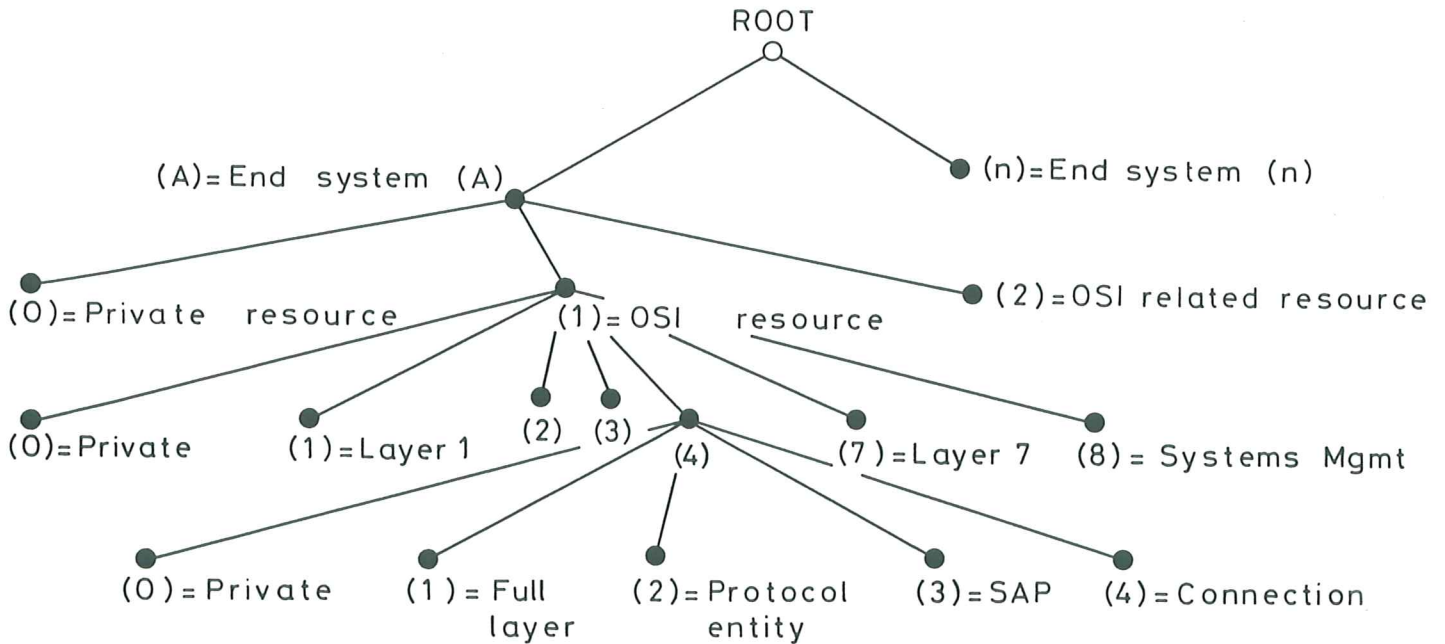


Figure A.1 - Management Information Tree for the Identification of Managed Objects

The name of an end system is the means by which the end system is identified. The values in parentheses are the specific encodings used to denote each entry within the tree. For managed objects representing OSI resources, it is suggested that the next most useful sub-entry is the OSI layer to which they relate. These are indicated simply by their layer number. Further resource types are identified for systems management itself and private use.

Note A.1

Private use is for further study.

Using this scheme the structure in the figure above has been built up by specific illustration with respect to the transport layer - a different structure may be required for each other layer, some of which may be markedly different from that described for the transport layer. This is for further study.

The specific denotations on the tree are those currently defined. Other values are for further study.

The distinguished name that uniquely identifies the specific SAP shown in the above Figure is a concatenation of the relative distinguished names of the superior entries in descending order, e.g.

A,1,4,3,<SAP identification >

The relative distinguished name of the SAP is:

3,<SAP identification >

A.2.3 Information Structure of Managed Objects

The information which is held in the MIB about managed objects is the representation of these objects for management purposes.

The collection of information about a managed object is called a Management Object Entry (MOE). The structure of MOE's is depicted in the following Figure:

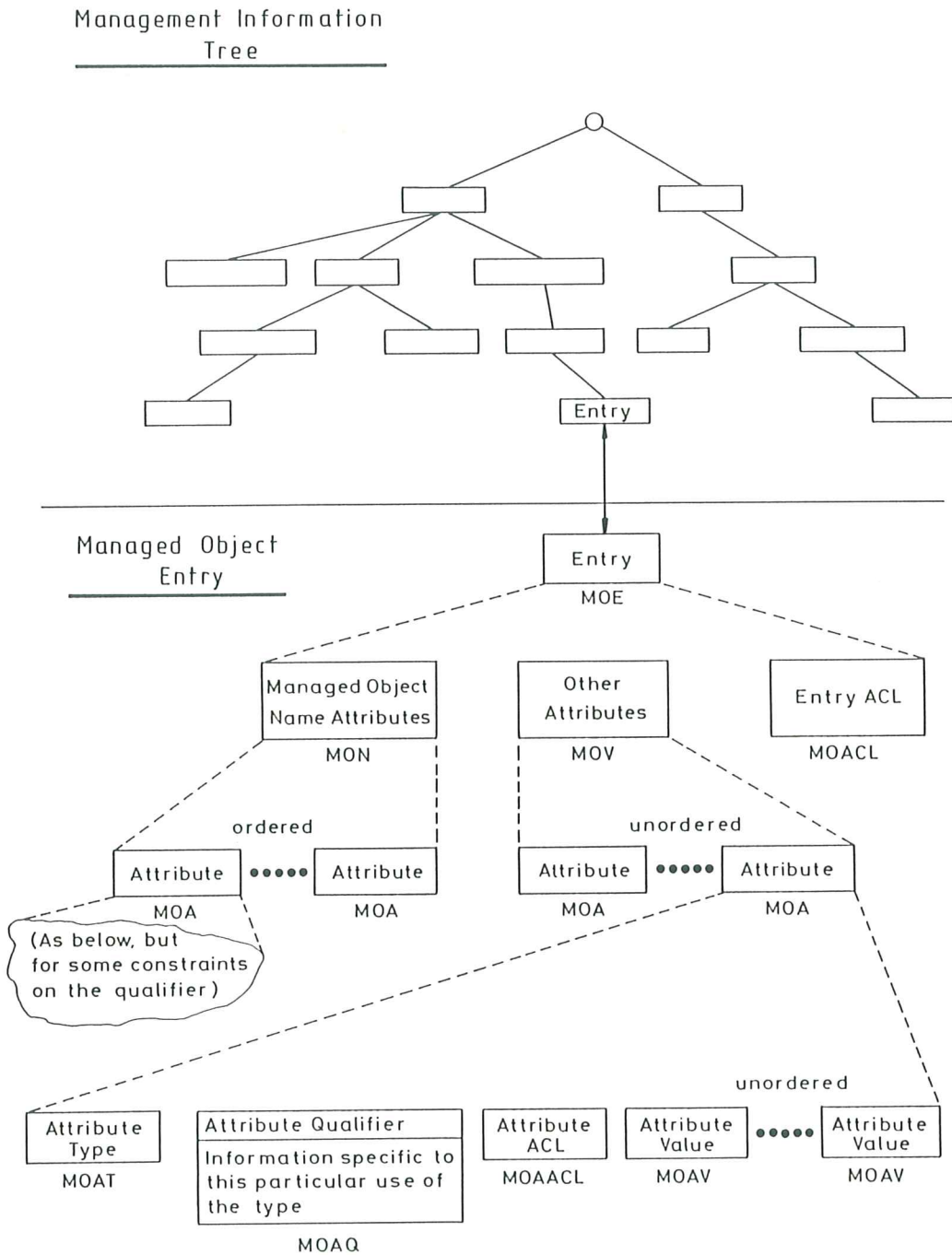


Figure A.2 - Structure of the Management Information Tree and of Entries

The **Management Object Entry (MOE)** can be further structured:

- **Management Object Name (MON)** is a set of information that identifies a management object entry.
- **Management Object Value (MOV)** is the value part of the MOE; it is composed of attributes.

- **Management Object Access Control List (MOACL)** is a set of information controlling the access to the MOE. It defines the permission and constraints which apply to this MOE, including those which apply by default to the attributes (for further study).

Management Object Class (MOC) is an attribute of a managed object representing a generic classification of similar objects and implicitly defines the structure of the MOE (i.e. which attribute types).

Management Object Attribute (MOA) is the information of a particular type concerning an object and appearing in a MOE describing the object in the MIB. An attribute has a number of component parts:

- **MOA Type (MOAT)** is a piece of information which identifies and classifies the specific attributes within an MOE.
- **MOA Qualifier (MOAQ)** is a piece of information which qualifies a particular attribute in the context of MOE. (The use of this is for further study.)
- **MOA Value (MOAV)** is a particular instance of the class of information indicated by an MOA type - there may be zero, one, or more occurrences in the attribute.
- **MOA Access Control List (MOAACL)** controls the access to the MOA, in the same way as the MOACL.

MOA Value Assertion (MOAVA) indicates a combination of an MOAT and MOAV, which asserts that within some entry (known or to be determined) there exists an attribute of that type one of whose values is that value.

MOE Relative Distinguished Name (MOERDN) is a property of a MOE which consists of either:

- a single MOAVA, or
- an ordered set of MOAVA's

MOE Distinguished Name (MOEDN) is a property of an MOE which is defined as the sequence of the MOERDNs of the MOE and those of all its superior entries in the MIT.

Management Information Tree (MIT) is the MIB considered as a tree, whose vertices (other than root) are MOEs.

Within the Common Management Information Protocol (CMIP) there is provision for the incorporation of a field called the resource Id.

It is in this field that the specific management object being referenced is placed. The reference is assembled by the concatenation of the individual value fields in the hierarchical structure, so as to form the name of the referenced object (using <x> <y> below). To this is concatenated the encoding of the specific attribute (<z> below). The marker < > is used in this denotation to signify the next element in the structure is the attribute. This is indicated in the examples below by the denotation

<x> <y> <z> etcetera

to the appropriate level.

Example 1:

In order to reference the attribute which contains the number of re-transmitted TPDU's for a specific protocol entity, using the above scheme, the following denotation would be employed:

<1> <4> <2> <10> <protocol entity identification>

Example 2:

In order to reference the status of a specific SAP, the following denotation would be employed:

<1> <4> <3> <100> <SAP identification>

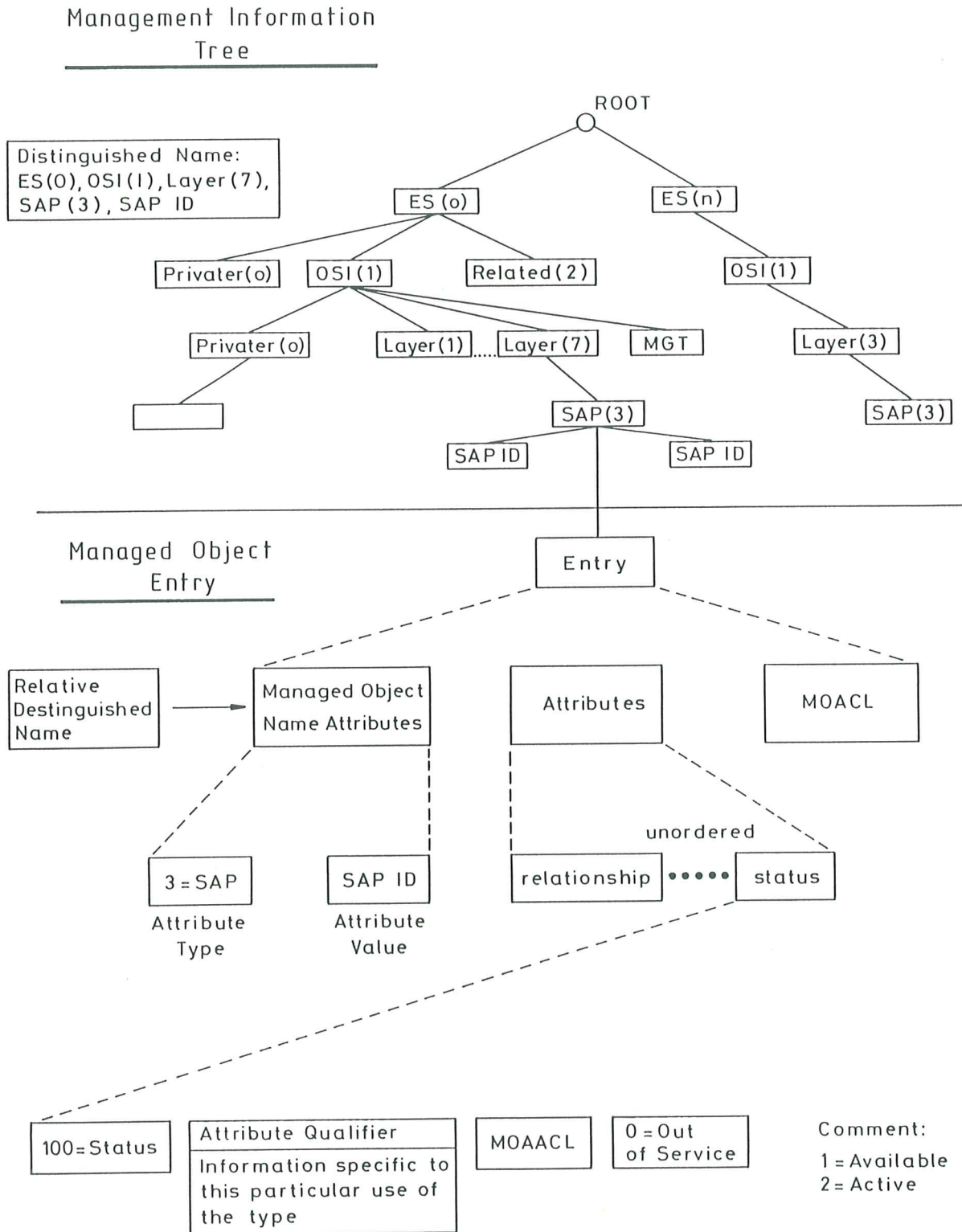
Example 3:

In order to reference the status of specific transport connection, the following denotation would be employed:

<1> <4> <100> <connection identification>

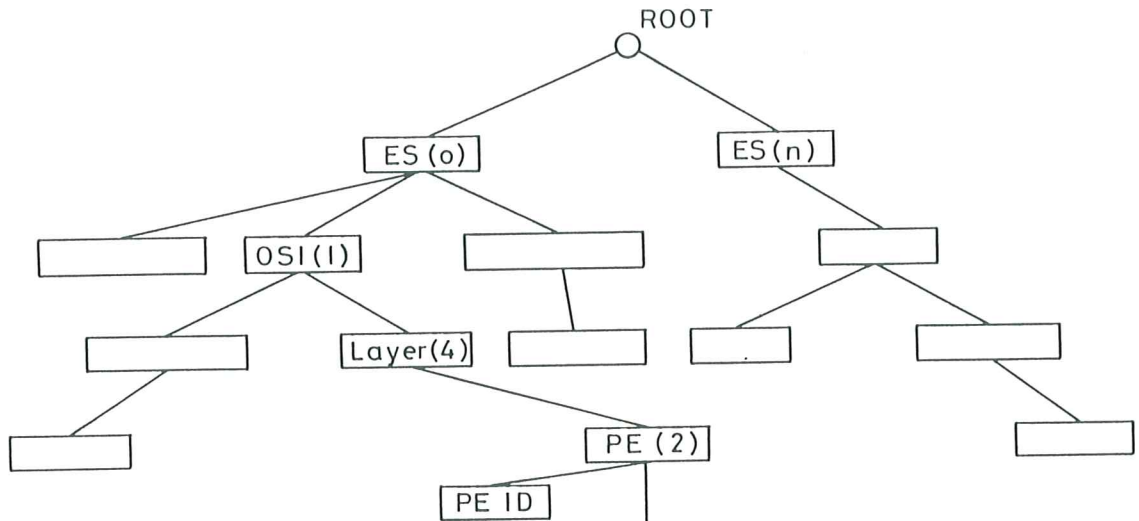
Structure of the Management Information Tree and of Entries

Example 1

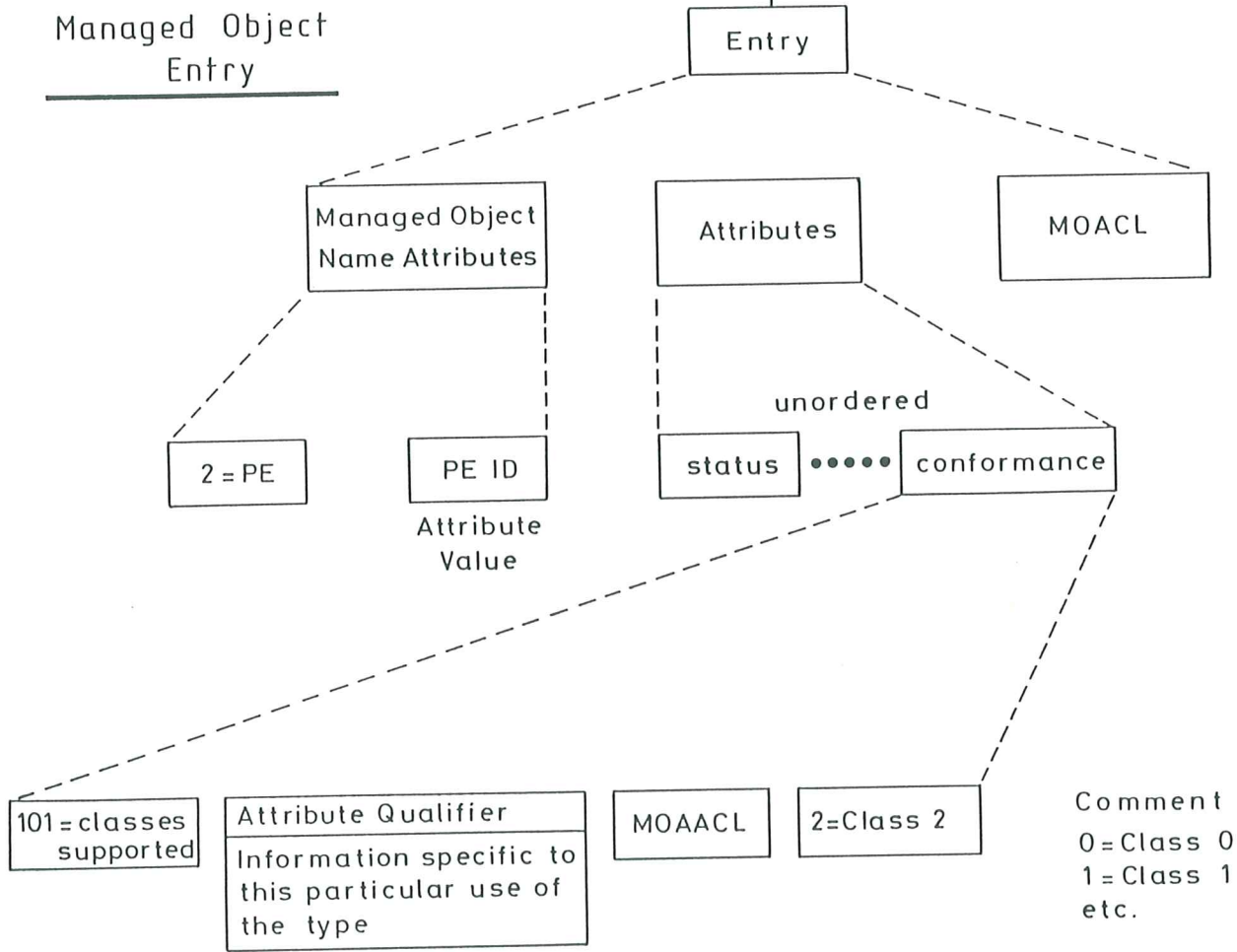


Example 2

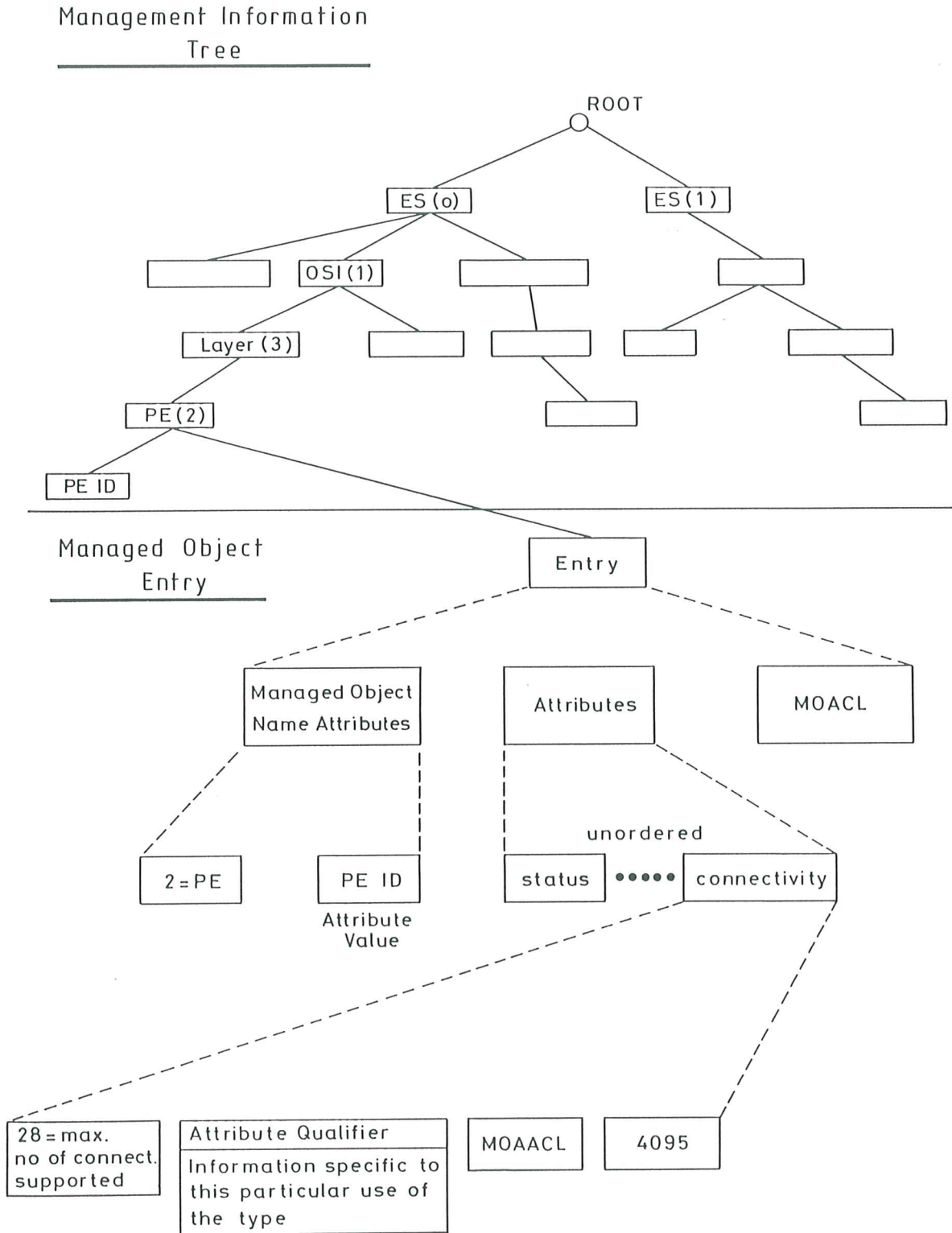
Management Information
Tree



Managed Object
Entry

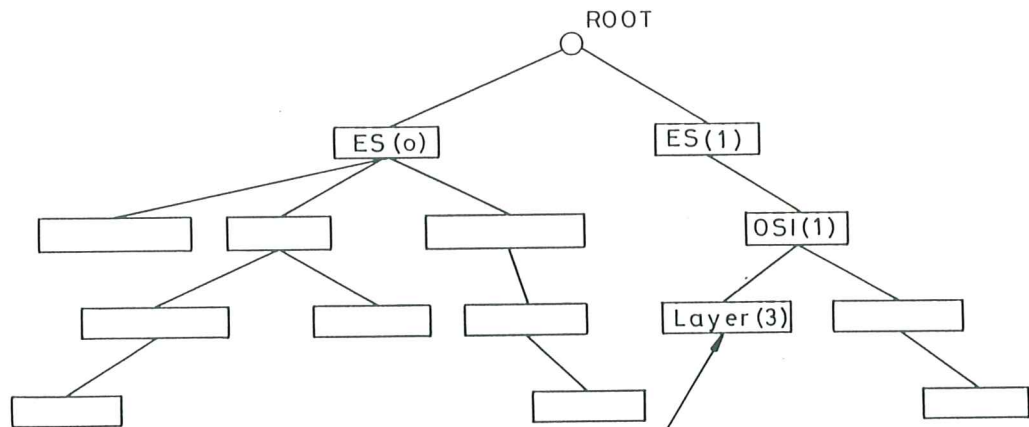


Example 3



Example 4

Management Information Tree



Managed Object Entry

