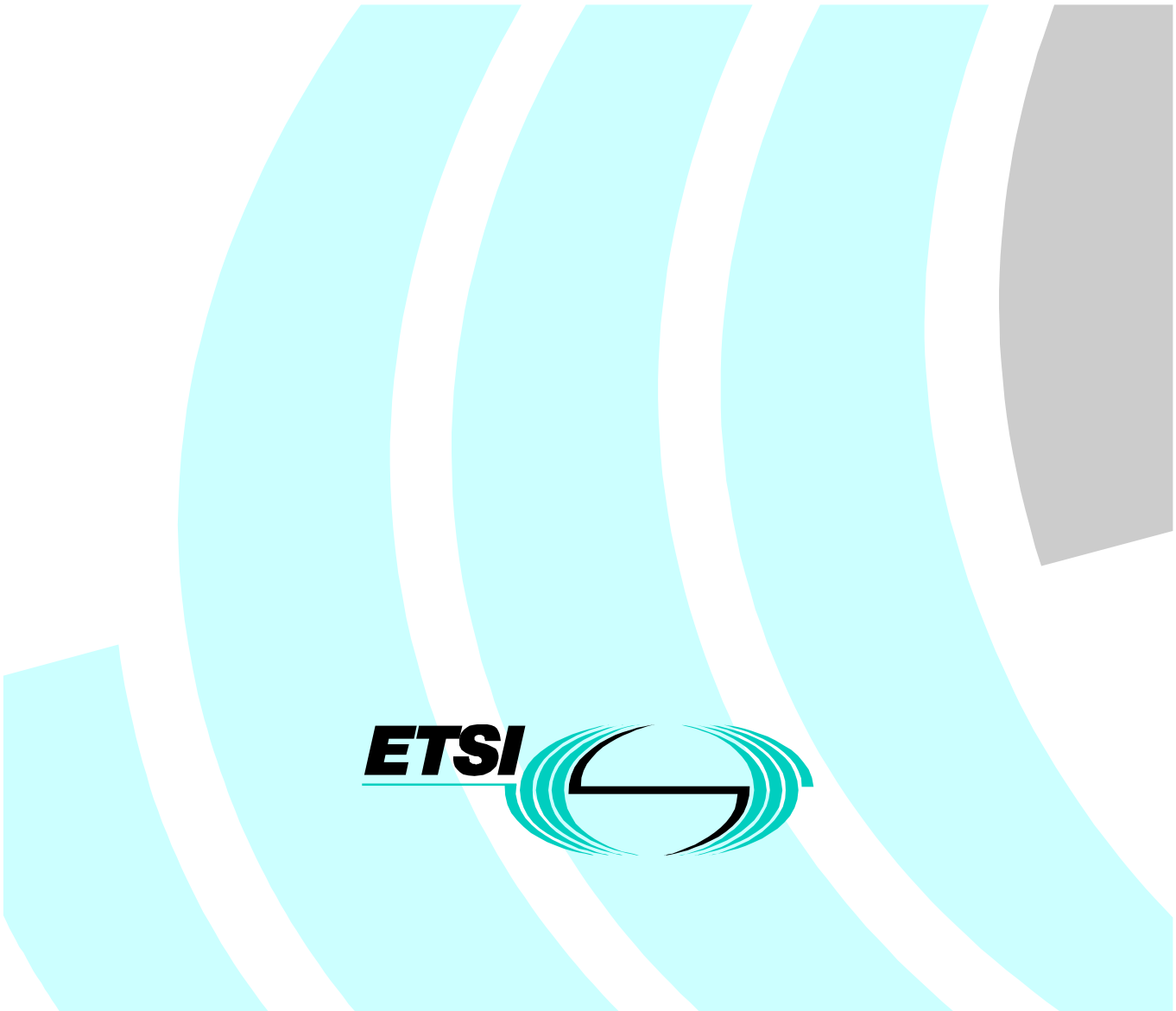
TR 101 648 V1.2.1 (1999-02)

Technical Report

Telecommunications Management Network (TMN); Managed object modelling guidelines



Reference
RTR/TMN-00029 (f9c00ios.PDF)

Keywords
TMN

ETSI

Postal address
F-06921 Sophia Antipolis Cedex - FRANCE

Office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16
Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Internet
secretariat@etsi.fr
Individual copies of this ETSI deliverable
can be downloaded from
http://www.etsi.org
If you find errors in the present document, send your comment
to: editor@etsi.fr

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available **free of charge** from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://www.etsi.org/ipr).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Telecommunications Management Network (TMN).

It is anticipated that the present document will continue to evolve as ETSI Working Groups gain experience in the specification of TMN interfaces. ETSI intends to review and update these guidelines on a regular basis.

1 Scope

The present document provides guidelines for the definition of additional Managed Object Classes, the use of existing Managed Object Class definitions and the inclusion of these objects in existing inheritance and containment hierarchies.

The present document:

- Provides general rules for defining Managed Objects.
- Provides general rules for defining Generic Object Classes.
- Provides additional Guidelines on the use of GDMO.
- Identifies relevant standards containing modelling guidelines, and identifies which clauses are recommended for use in ETSI models (for example where options exist).

The present document does not:

- Imply any specific system implementation.
- Include testing the Information model.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.
- A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

[1] ITU-T Recommendation M.3100: "Generic Network information Model (1st revision)".

[2] ITU-T Recommendation M.3020: "TMN interface specification methodology".

[3] Void.

[4] ITU-T Recommendation X.700: "Management framework for Open Systems Interconnection (OSI) for CCITT applications".

[5] ITU-T Recommendation X.701: "Information technology – Open Systems Interconnection – Systems management overview".

[6] ITU-T Recommendation X.720: "Information technology – Open Systems Interconnection – Structure of management information: Management information model".

[7] ITU-T Recommendation X.721: "Information technology – Open Systems Interconnection – Structure of management information: Definition of management information".

[8] ITU-T Recommendation X.722: "Information technology – Open Systems Interconnection – Structure of Management Information: Guidelines for the definition of managed objects".

[9] Void.

[10] Void.

[11] Void.

[12] ETR 230: "Network Aspects (NA); Telecommunications Management Network (TMN); TMN standardisation overview".

[13] ITU-T Recommendation M.3010: "Principles for a Telecommunications management network".

[14] ISO/IEC 10165-1: "Information technology - Open Systems Interconnection - Structure of management information: Management Information Model".

[15] ISO/IEC 10165-2: "Information technology - Open Systems Interconnection - Structure of management information: Definition of management information".

[16] ITU-T Recommendation X.710: Information technology – Open Systems Interconnection – Common Management Information Service

[17] ITU-T Recommendation G.851-1: "Management of the transport network – Application of the RM-ODP framework".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

standards: this generic term is used in the present document to refer to ITU-R Recommendations, ETSI ETSs ENs and ETRs, ECMA TRs, ISO/IEC standards and EWOS TRs.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ASN.1	Abstract Syntax Notation One
CMIP	Common Management Information Protocol
CMIS	Common Management Information Service
DMI	Definition of Management Information
ECMA	European Computer Manufacturers Association
E/R	Entity-Relationship
ETR	ETSI Technical Report
ETS	European Telecommunication Standard
ETSI	European Telecommunications Standards Institute
EURESCOM	European Institute for Research and Strategic Studies in Telecommunications
EWOS	European Workshop for Open Systems
GDMO	Guidelines for the Definition of Managed Objects
IDL	Interface Description Language
IEC	International Electro-technical Commission
ISO	International Organization for Standardization
ITU	International Telecommunication Union
ITU-T	ITU Telecommunications sector
LLA	Logical layered Architecture
MF	Management Function
MFS	Management Function Set
MIM	Management Information Model
MO	Managed Object
MOCS	Managed Object Conformance Statement
MS	Management Service
NMF	Network Management Forum
OMG	Object Management Group
OMT	Object Modelling Technique
OSI	Open Systems Interconnection
PICS	Protocol Implementation Conformance Statement

PNO	Public Network Operator
RM-ODP	Reference Model for Open Distributed Processing
SMF	Systems Management Function
SMI	Structure of Management Information
SNMP	Simple Network Management Protocol
TMN	Telecommunications Management Network
TR	Technical Report

4 Purpose and Structure of Modelling Guidelines

The objective of the present document is to provide guidance to those ETSI Working Groups producing TMN Management Information Models (MIM).

It aims to:

- enable models developed by different groups to co-exist;
- assist in producing consistency across models developed by different groups.

The use of the guidelines is intended to produce Managed Object (MO) models that are implementable and reusable. It is therefore important to reuse the work of others and develop models in such a way that they can themselves be reused.

It is intended that these modelling guidelines should be "Methodology Neutral".

These guidelines assume that whenever possible and appropriate, the ITU-T Recommendation X.700 [4] Systems Management or ODMA Recommendations will be used as a basis of TMN managed object modelling. An overview of the relevant ITU-T Recommendation X.700 [4] can be found in ETR 230 [12] TMN Standardization Overview.

The guidelines start by considering methods for requirements capture, they go on to consider Information and Dynamic Aspects of information modelling. The use of the NMF ensemble approach for documenting information models is considered, together with the use of Class libraries. This is followed by a consideration of principles for GDMO implementation. Consideration of other protocols to be used in the evolving TMN is for further study.

Testing the information model is considered to be outside the scope of these guidelines.

5 Pre-Information Modelling Work

5.1 Introduction

The development of Management Information Models is just one part of the activities involved in the definition and development of TMNs. There are several models of software system (waterfall, spiral) which distinguish between different activities. There is a consensus that eliciting and analysis of the needs and constraints expressed by the procurers of the system is part of such activities. This activity can have names such are requirement capture, requirement definition to name but a few. If we consider management information modelling to be a specification/design activity it is natural to think that requirements related activity precedes it. This clause briefly discusses the requirement related activities which typically will take place, in part, at least prior to management information modelling in a TMN context.

The origins of requirements on TMN systems that drive the definition of information models are wide and varied. It is not appropriate to address such philosophical matters here. It is assumed that, at a point in the development of TMN systems or standards, the scope of a management problem is defined at a sufficient level of detail to allow the modelling of management information as a set of managed objects.

Information about a management problem scope can be organized in a number of ways which provide input to the management information modelling activity. These include:

- Definition of Management Services, etc. following ITU-T Recommendation M.3020 [2].
- Definition of Management Functions, etc. following the Ensemble concept defined by NMF.
- Application of Open Distributed Processing (ODP) system view-points following ITU-T Recommendation G.851-1 [17].

All of these approaches have a number of features in common. The person modelling defines:

- Entities (Resources) that will be responsive to requests to engage in their management.
- Functions of people and machines which request entities to engage in management activity to achieve particular management goals.

Typical entities are telephone exchange switches, cables for the transport of telephone traffic, counters representing the volume of traffic carried by a particular circuit, etc.

It has been popular to categorize functions into those for configuration management, fault management, performance management, security management and accounting management.

Management goals are states of the entities that the owners of the entities wish to establish and maintain. For example, a PNO running a particular network requires network equipment mis-operations to be unambiguously identified or a service subscriber wants to be notified of the duration of a service interruption affecting it.

Before tackling a management information modelling task, available requirement statements should be examined to detect:

- Management goals.
- Management functions.
- Managed resources.

During this examination, it is advisable to organize the information in an orderly manner according to the approaches listed above, each of which is defined briefly as follows.

5.2 M.3020 TMN Interface Definition Methodology

5.2.1 General

The TMN interface specification methodology is defined in ITU-T Recommendation M.3020 [2]. This Recommendation describes the process of deriving TMN interface specifications from TMN user requirements. The process is divided into a number of steps.

User requirements are expressed as management services, which are decomposed into management functions. During the detailed specification activities, these management functions are directly mapped to protocol specific messages; in the case of CMIP, to operations on managed objects and notifications from managed objects (see clause 6). The initial steps of the process cover this part of the methodology. The latter steps cover the protocol stack definition part of the methodology; protocol stacks are outside the scope of the present document.

5.2.2 The definition process

5.2.2.1 Tasks, information and relationships

Several tasks are linked to form the TMN interface specification process, as defined in ITU-T Recommendation M.3020 [2]. These tasks are defined as follows (excluding protocol tasks):

- describe TMN management services;
- describe TMN management context;

- perform information modelling;
- consolidate available information;
- define management information schema.

Each of these tasks involves the production or use of specific information. Figure 1 illustrates the process, showing the relationships between the tasks and types of information. (This figure is based on figure 10 in the ITU-T Recommendation M.3020 [2]).

5.2.2.2 Describe TMN management services task

This task is concerned with identifying the management activities to be supported by the TMN. For each activity the management goals and examples of the benefits to TMN users from these goals are identified.

5.2.2.3 Describe TMN management context task

This task is concerned with identifying the roles, resources and functions associated with each management activity. Relationships are also specified, in the form of scenarios.

5.2.2.4 Perform information modelling task

This task is concerned with identifying existing (generic and technology-specific) and new object classes which will support the management activities identified earlier in the process.

5.2.2.5 Consolidate available information task

This task is concerned with ensuring that each and every management function is supported by one or more object classes. This may require the previous two tasks to be repeated iteratively until the set of object classes is satisfactory.

5.2.2.6 Define management information schema task

This task is concerned with defining the information model in terms of the managing system view of the managed system. In particular, it identifies the containment (naming) relationships for the object classes. The output of this task is a complete specification of the information exchange aspects of a TMN interface.

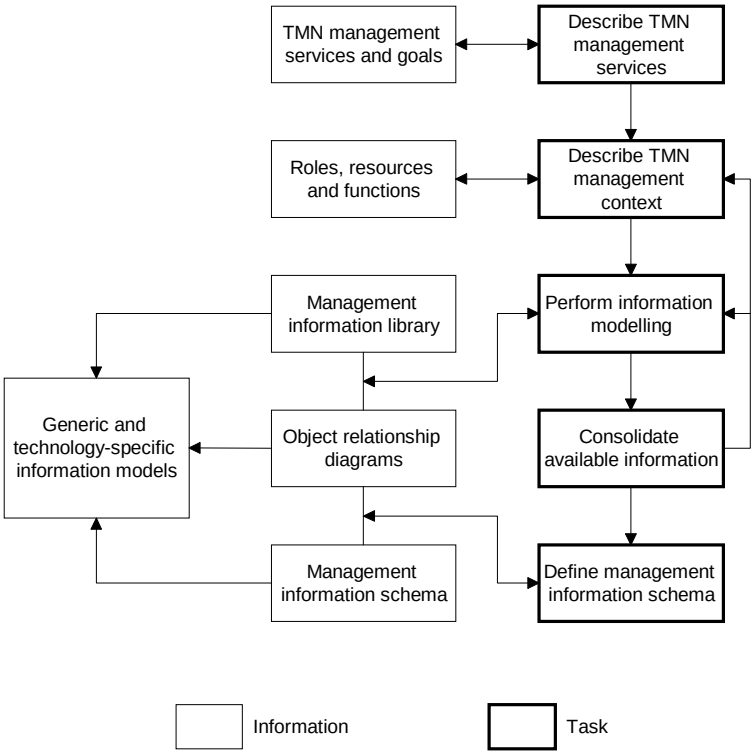


Figure 1: TMN interface specification process and information

5.3 The Ensemble - a specification framework

5.3.1 General

The Ensemble concept was developed by the Network Management Forum (NMF) "[…] to enable management system specifiers to define clearly which management standards are appropriate to their problems and how they are to be used". In other words, an Ensemble is a complete specification of a management problem and its solution.

The Ensemble provides the TMN interface specifier with a framework for combining the products of a TMN standards-based specification development into a cohesive unit.

The NMF has also developed a concept called the Solution Set [NMF Man.]. A Solution Set is defined as a "[…] package of standards and specifications that provide all the detail necessary to implement an NMF agreement" [NMF Man]. They are intended to be extremely precise because they apply to a single implementation environment and do not permit options. (Unfortunately, the latter statement is contradicted later in the document). Any two management systems that implement a Solution Set should interoperate.

A Solution Set contains a Product Descriptor and one or more Ensembles. The Product Descriptor summarizes the business problem being solved, the scope of the agreement, the benefits to the target audience and its applicability in different business contexts.

The Ensemble is therefore a key component of a Solution Set. However, the Solution Set concept is very new and few such specifications have been produced. The intention of including multiple Ensembles in a Solution Set is that the Ensembles differ from each other in terms of the protocol to be used (CMIP, SNMP, etc.). It is unclear at this stage how this concept should be applied, apart from within the NMF.

5.3.2 Ensemble structure

The basic structure of an Ensemble is made up of four main sections, which may be further enhanced and extended by the specifier:

Introduction, containing general header information, a brief description of the Ensemble, its scope and any relationships to other Ensembles.

Management context, containing detailed information on the resources and functionality (the management problem) supported by the Ensemble, including the management viewpoint encompassing the management problem.

Information model, detailing the managed object classes representing the resources and functionality supported by the Ensemble. Class and instance relationships are captured in suitable diagrammatic form. A set of scenarios describing how the functionality is applied to the resources is also specified. The original intention was that the Ensemble would not contain managed object class definitions, but refer to classes in a specific library of definitions.

Conformance requirements, containing general conformance requirements, functional support, MOCS proformas and associated CMIS services support tables.

A template for Ensembles, including section headings and common text, is provided in "The Ensemble Concepts and Format" document.

Since the *Introduction* section is general in nature, it is not discussed in the present document.

5.3.2.1 Management context

5.3.2.1.1 Management view and level of abstraction

The management view and level of abstraction may be defined in general terms using the TMN LLA ITU-T Recommendation M.3010 [13] and the OSI systems management functional areas ITU-T Recommendation X.701 [5]. The layers of the TMN LLA can be placed on one axis of a table, and the functional areas on the other axis. This approach is illustrated in table 1 below. The example Ensemble is concerned with aspects of configuration management in the network management layer.

Table 1: Identification of general Ensemble management view and level of abstraction

	Fault	Configuration	Accounting	Performance	Security	Other (note)
Business Management						
Service Management						
Network Management		✔				
Element Management						
NOTE: A new functional area is defined where the existing functional areas are inappropriate.						

It is then necessary to identify the entities to which the Ensemble applies and the roles those entities take in communicating via the specified interface.

The TMN architecture ITU-T Recommendation M.3010 [13] is used to classify the entities and determine the type of interface between them. The roles are determined by how the systems management model ITU-T Recommendation X.701 [5] is best applied to the particular management problem being addressed.

Assume that a network management system is required to support the necessary functionality to provide VP configuration for an ATM cross-connect network. The cross-connect switches in the ATM network are all managed by a single switch management system. See figure 2.

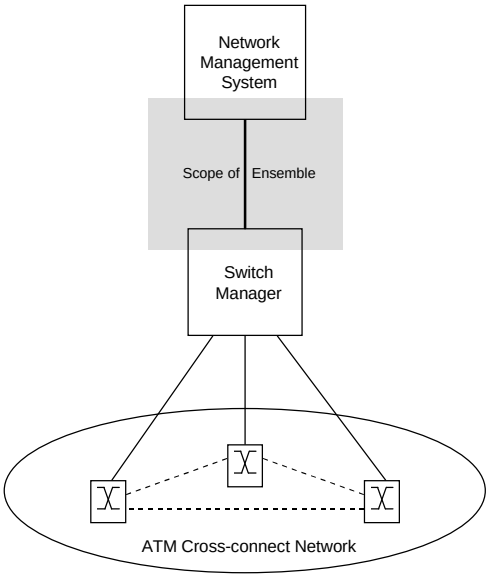


Figure 2: Example of management system

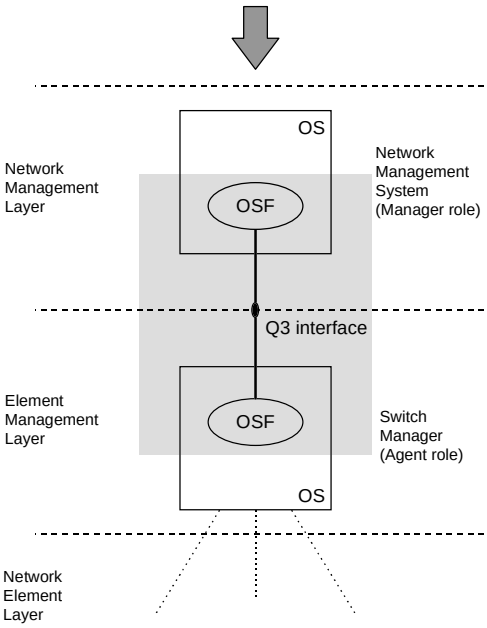


Figure 3: Example illustrating management view, entities and roles

The TMN interface has now been identified, with Manager and Agent roles assigned to the communicating entities. (See figure 3) The scope of the Ensemble has been constrained in part. It is now necessary to specify precisely which resources are to be managed, and what management functions are to be supported.

5.3.2.1.2 Resources

The resources are those entities, physical or logical, which are to be managed. Only those resources which are relevant to the Ensemble are described. The descriptions must be sufficient to enable the specifier to identify appropriate managed object classes and their relationships. The greater the detail available, the more accurately the specifier may define the contents (data structures) of these classes. It should be noted that there does not necessarily exist a one-to-one relationship between individual resources and managed object classes.

Physical resources are usually network equipment or systems, for example multiplexers, switches or service platforms. Logical resources are either abstractions of physical resources (above the Element Management Layer in the TMN LLA ITU-T Recommendation M.3010 [13]), or support functions, for example logs and records, or event forwarding discriminators ITU-T Recommendation X.721 [7].

5.3.2.1.3 Functions

The management functions are defined at three levels ITU-T Recommendation M.3020 [2]:

- Management Services (MSs).
- Management Function Sets (MFSs).
- Management Functions (MFs).

These levels provide a convenient means of categorizing and decomposing Ensemble functionality. The MFs themselves map directly to CMIS services ITU-T Recommendation X.710 [16]. It is therefore possible to directly relate MFs to managed object class attribute operations, actions and notifications.

The specification of MSs, MFSs and MFs should include unique names/identifiers, descriptions, their inter-relationships and identification of the resources to which they apply.

Existing definitions may be re-used. In these cases, reference documents should be stated, and any restrictions or constraints on the original definitions specified.

5.3.2.2 Information model

Information model specification is addressed in clause 6 of the present document.

5.3.2.3 Scenarios

Scenarios define the dynamic aspects of the Ensemble specification, showing how the managed object classes are used to support the functions defined in the management context. All the scenarios relevant to the Ensemble must be defined, all functions must map to a scenario.

Scenarios are defined using message flow diagrams and text descriptions. The message flow diagrams consist of CMIS ITU-T Recommendation X.710 [16] request/response primitive pairs (or an unconfirmed CMIS ITU-T Recommendation X.710 [16] request primitive) exchanged between Manager and Agent systems ITU-T Recommendation X.701 [5], which map directly to a function. The general format for a message flow diagram is shown in figure 3a.

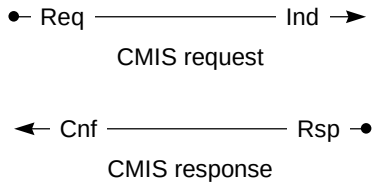


Figure 3a: General format for message flow diagram

5.4 ODP based approach (ITU-T G.851-1) - Methodology overview

5.4.1 Introduction

From a network management point of view, the RM-ODP (Reference Model for Open Distributed Processing) provides an object oriented framework (for distributed management systems) that allows user requirements for each management application (for example configuration management), the information (or data) related to the resources to be managed and the way that the information may be accessed and manipulated to be defined in a way that is essentially independent of the technology and distribution used in the implementation of a management system.

The RM-ODP framework provides five viewpoints of the system and resources being managed. This is illustrated in figure 4.

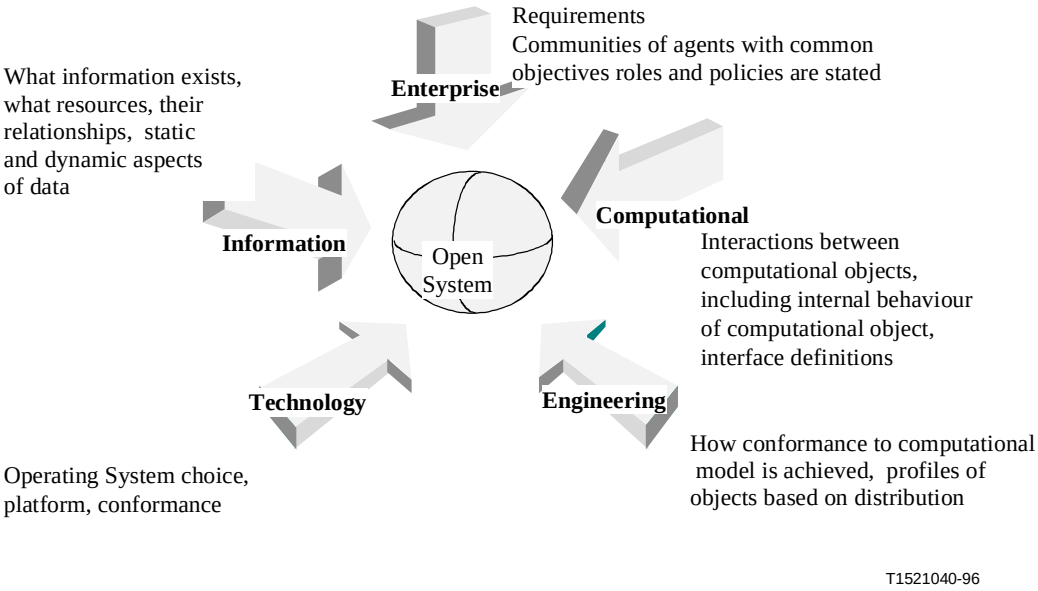


Figure 4: RM-ODP viewpoints (ITU-T Recommendation G.851-1)

The RM-ODP provides:

- a set of concepts to describe open distributed systems;
- independence with regard to any existing analysis and/or design and/or programming method/language.

The initial network level modelling work uses only the enterprise, information, computational and engineering viewpoints from RM-ODP. The technology viewpoint is not currently considered to be within the scope of the Telecommunication Standardization Sector of the ITU.

The information and computational viewpoints of the network level model focus on the semantics rather than the syntax of the information that must be exchanged for each individual or set of management applications. This level of definition is sufficient to allow more detailed specification of the system in the engineering viewpoint where the interoperability of the system is defined by the external interfaces. Conformance requirements will typically be defined in relation to these external interfaces. The engineering viewpoint shall be defined for a specific communications technology, such as OSI systems management.

This methodology document defines the viewpoints used to define the network level model.

5.4.2 RM-ODP viewpoint descriptions

Complete descriptions of the RM-ODP viewpoints can be found in the references listed in clause 2. A brief description of RM-ODP viewpoints follows:

- Enterprise viewpoint: A viewpoint on an ODP system and its environment that focuses on the purpose, scope and policies of that system. An enterprise specification enables the client of an ODP system to express its requirements and policies thereby establishing a contract between the client and the provider of the ODP system. Thus, an enterprise specification must be expressed in terms that are understandable by both parties.
- Information viewpoint: A viewpoint on an ODP system and its environment that focuses on the semantics of information and information processing activities in that system. An information viewpoint focuses on the definition of information object types, together with their relationships, their state values and permitted state changes.
- Computational viewpoint: A viewpoint on an ODP system and its environment that focuses on functional decomposition into structures suitable for distribution. A computational specification describes computational object types and their interface types. Instances of these object types will cooperate with each other through interfaces. Operational interface types are defined by specifying their operation signatures and associated behaviour. All computational objects are potentially distributed into separate systems. In the context of this methodology, any computational objects that are defined cannot be further distributed unless further computational interfaces are defined.

- Engineering viewpoint: A viewpoint on an ODP system and its environment that focuses on the functions required to support distribution in that system. An engineering specification is used to make actual decisions regarding distribution of the computational objects and, in addition, determine the infrastructure components required to support the distributed objects.

- Technology viewpoint: A viewpoint on an ODP system and its environment that focuses on the choice of technology in that system.

5.4.3 Using RM-ODP viewpoints in a specification design methodology

Although the reference model for open distributed processing provides an approach to modelling (i.e. the RM-ODP viewpoints), it does not provide a prescriptive methodology that can be followed in developing a system. This subclause introduces the methodology, using the RM-ODP viewpoints, for use in the standardization of transport network management models, whereby the application components are viewed as objects communicating through well-defined interfaces, and for which only the externally observable behaviour is described in an implementation-independent manner. The methodology uses the following steps:

- requirements are identified;

- information to describe the system is defined;

- processes that manipulate the information and provide the services are described; and

- distribution and implementation decisions are made.

Each of these design steps is associated with an RM-ODP viewpoint. In practice the temporal ordering of the viewpoints is not required. What is more important is the separation of concerns provided by the RM-ODP framework.

The set of resources that form a transport network architecture can be described using the RM-ODP enterprise and information viewpoints only, whereas a network management service requires the addition of the computational viewpoint. Both the transport network architecture and service definition will be merged during the engineering process to develop management capabilities at, for example, a Q3 interface.

5.5 Additional Information

This subclause collects some advice which augments the methodologies described above.

5.5.1 Representation of structure

Rather than trying to represent all aspects of the structure of the managed resources, strive for the simplest representation of the structure which just meets the management requirements.

Use attributes within a single object to model parts of resource structure provided that:

- they are intuitively passive data rather than active functions;

- the structure to be modelled has no more than three levels of depth;

- neither ordering nor repetition of attributes is required; and

- the number of attributes in each instance is "reasonably small".

5.5.2 Goals of the pre-information modelling work

In summary, without the expenditure of an enormous effort, requirements should be:

- unambiguous;

- stated in a structured format;

- clear and accessible to a non-expert reader; and
- non overlapping.

It should be possible to trace through from the requirements to the solution (management interface specification).

5.5.3 Reference material

Additional information supporting the definition of management requirements and information can be found at the EURESCOM Website called E-MOL ([WWW]: http://www.eurescom.de/~public-web-deliverables/p600-series/P609/emolgate.htm). This site contains repositories of information likely to be useful to people defining information models, ensembles and management services. One of the repositories is devoted to TMN guidelines.

6 Principles for the application of GDMO

6.1 Introduction

The preceding clause showed only the first step of the process solving a particular TMN problem. In the second step, a formal or semi-formal notation must be applied to create the entities of the information model describing a TMN system. This clause provides guidelines on how to create object classes using the GDMO syntax. It is intended to be a short, useful and practical introduction to the pertaining ITU-T Recommendation X.700 [4] series on information modelling and GDMO.

Given a small number of restrictions concerning the use of the more complex GDMO constructs and the scale of the implementations, the OSI domain - represented by GDMO - can perfectly co-exist with the OMG domain - represented by IDL. On the translation between GDMO and IDL, the reader is referred to the work of JIDM (a joint X/Open and NMF initiative) or to E-MOL's advanced guidelines (http://www.eurescom.de/~public-web-deliverables/p600series/P609/eurescom/emol97/guidelines-lib/specification/gdmo_lite/index.html).

6.2 Use of Attributes

6.2.1 Value Set Specification for Attributes

In general only the basic ASN.1 Type should be specified in an attribute template using ATTRIBUTE SYNTAX. Do not use subtyping or PERMITTED VALUES syntax. The basic type of an attribute is fixed under inheritance and can never change.

Required Values are those values that must be fully supported by an instantiated managed object. This is the minimum set of values supported by an object. Required values are ASN.1 types which include some or all of the values in the corresponding base types. The set of required values may be extended under inheritance, but cannot be restricted.

Permitted values are those values that may be fully supported by a specific system. They are a subset of the basic type and a superset or the required values. The permitted values set is the maximum set of values supported by a managed object class, in order for that managed object class to be conformant. The set of permitted values may be restricted under inheritance, but cannot be extended.

Supported Values are those values of the attribute that a particular managed object supports in a particular agent system. This set of values is not specified in a managed object class, but must be specified in the Managed Object Conformance Statement (MOCS). The Supported values set is a subset of the Permitted Values set and a superset of the Required values set.

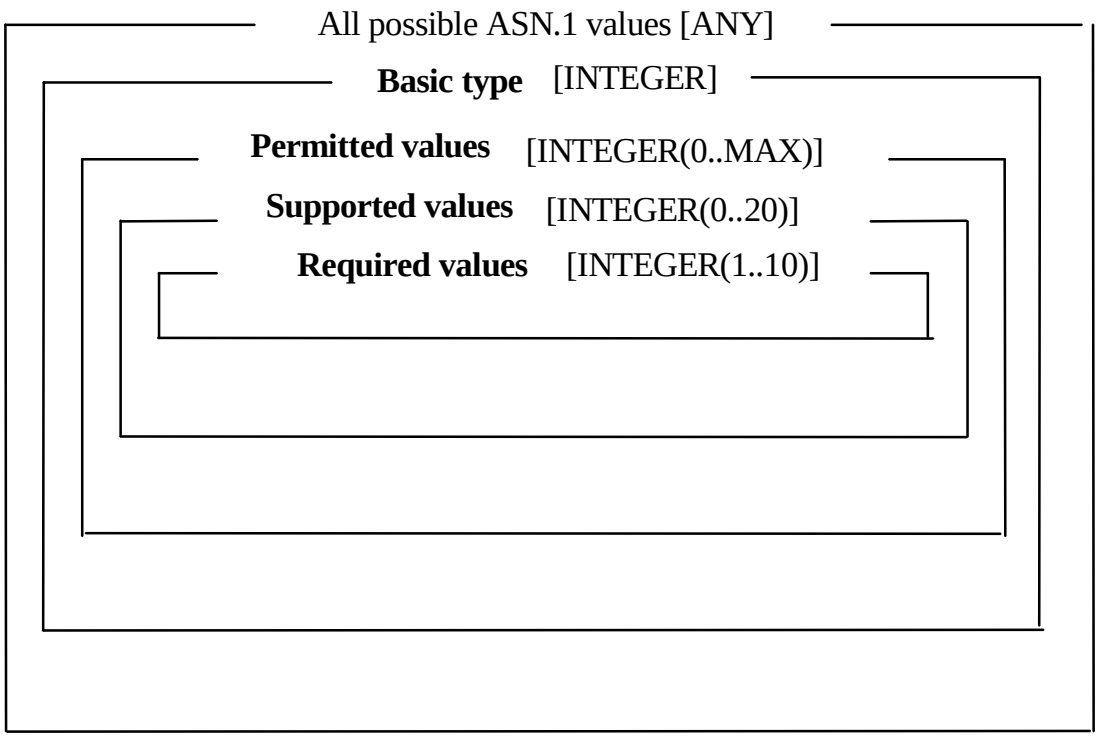


Figure 5: Sets of Attribute Values

6.2.2 Specifying Required, Permitted and Supported Values

The permitted and required values sets may be included in the specification of the managed object class when the attribute is inserted into that class, or when the attribute is inherited by a subclass. These sets could be further modified (subject to the same rules) in an implementation agreement. This option could be used in cases where the set is specified for the sake of achieving interoperability, and it is necessary to change the set over time without affecting the registered class specifications.

As noted above, the supported values set is not a part of any attribute or class specification, but is stated by an implementor in the appropriate MOCS.

MOCS proformas should not be used to specify permitted or required value sets, since this practice tends to hide the agreement.

6.2.3 Guidelines for Attribute-Value Set Selection

When selecting the basic type (as defined above) for an attribute, choose the ASN.1 type that includes all values that could possibly fit within the semantics of the attribute. Specify this basic type in the actual definition of the attribute.

If the attribute is to be used in several different managed object classes where restrictions to the permitted value set or extensions to the required value set are anticipated to vary, define a managed object class (an "unrestricted superclass") with default permitted and required value sets, and then modify these sets as needed in its subclasses.

Use the ENUMERATED ASN.1 type only when it is not required or desirable to modify the required or supported value sets by adding new enumerations in subclasses.

To modify required or supported "enumerated" value sets in subclasses, you must use the INTEGER ASN.1 type with a NamedNumberList specification in the attribute specification. This approach defines the base types as INTEGER, regardless of the values in the NamedNumberList (this is used only for value notation), which permits modification of value sets in subclasses. In addition, new values can be added to the list in a subclass (to provide semantics for these values) without changing the actual value sets.

Extending an Operation or Notification Parameter List

Extensible parameter lists are extremely useful for (1) modifying an operation or notification that is inherited into a subclass from a superclass, (2) unifying two or more actions or notifications that are multiply inherited into a subclass, or (3) constructing allomorphic superclasses.

In general, if one of these three uses is contemplated when constructing operations or notifications for inclusion in an object class, then the operation or notification must have a constructor type specified as the single argument; i.e. a parameter list. Some useful constructor type are SEQUENCE OF or SET OF. If these are chosen, then the parameter list is extensible if the last parameter is declared to be a **Management Extension** type.

6.2.4 General Attribute Guidelines

When it is necessary to specify attributes in objects that are being developed in other experts groups, the group requesting the attribute must specify default values and behaviour.

Regional options should be contained in ITU objects as conditional packages so that they are available to other regions.

Group Attributes

Define group attributes for semantically related groups of attributes in cases where it is logical and convenient to perform get or replace-with-default operations on the group as a whole.

6.3 Use of Inheritance

6.3.1 General Inheritance Guidelines

Inheritance Rules

The inheritance rules defined in the present document shall not conflict with those specified in ITU-T Recommendation X.721 [7] Management Information Model and ITU-T Recommendation X.722 [8] Guidelines for the Definition of Managed Objects.

Conditional Packages are always inherited from the superclass. However the condition for the presence of the package in an instance of subclass is defined to be the logical OR of all inherited conditions for that package, together with the condition stated in the subclass. In particular, a conditional package can be made mandatory for all instances of the subclass (and any subclasses derived from the associated managed object class) by stating a condition of TRUE for the package in the subclass definition.

6.3.2 Principles and Guidelines for Instantiable Classes

Inheritance Rules

Specify a class as instantiable whenever any of the following conditions are met:

- The class models one or more aspects of a resource that need to be managed.
- The class is necessary to support management functions such as event logs and event reporting filters.

Instantiable object classes may be created by subclassing from Top (the top-level class), by inheriting from another uninstantiable class, or by inheriting from other instantiable managed object classes.

Instantiable classes should be defined through inheritance from uninstantiable or instantiable superclasses to model additional aspects of a resource that must be managed. This includes vendor specific extensions of a particular standard object class.

6.3.3 Principles and Guidelines for Uninstantiable Classes

Generate an uninstantiable class to group characteristics (attributes, operations and notifications) that will perform a similar function or purpose in a variety of managed objects. The variety of managed objects can then be defined as subclasses of the uninstantiated class.

Generate an uninstantiable class to ensure that all subclasses of a particular type (for example Event records) inherit the same characteristics and can therefore be managed in the same manner.

Uninstantiable classes should be used to facilitate reuse of specification. In a case where it is expected that a group of characteristics will be used to create many subclasses, where the characteristics are self-consistent, and where they can be documented independently of the classes into which they are to be inherited; the group of characteristics should be defined in an uninstantiable class.

Uninstantiable classes should be defined as high in the class hierarchy as possible in order to foster reuse of specification.

Do not use uninstantiable classes to model optionality or conditionality. Other, more appropriate mechanisms are available to perform this function. Nor should uninstantiable object classes be used to model resource structure.

There will be cases where it is expedient to define managed object classes that are not instantiable, but exist for the sole purpose of creating new classes via inheritance into subclasses. Using such an approach reduces specification and increases reuse of specification. The following are valid reasons for creating these uninstantiable object classes:

- to group characteristics that are related by a purpose such as resource function; and
- to ensure that all subclasses of a particular class have certain characteristics.

Use inheritance to model the IS-A relationship or for purposes of class extension or for both (as in multiple inheritance), even when dealing with uninstantiable classes. However, a component relationship, or IS-PART-OF relationship, should not be modelled with inheritance. Instead, use other methods, such as containment, or by including the characteristics directly in the class specification.

Consider creating an uninstantiable managed object class for a group of characteristics if the following constraints are satisfied:

- The group of characteristics are internally self-consistent, yet do not, by themselves, merit definition in an instantiable object class.
- They will or could be used to create many subclasses.
- Their syntax and semantics can be documented independently of the classes into which they are to be inherited.

6.4 Use of Multiple Inheritance

The principles and guidelines summarized in this clause extend, and sometimes supersede, those dealing with strict inheritance and described in the preceding clause. When specifying a managed object class using multiple inheritance, refer to the rules and guidelines for strict inheritance before applying those described in this clause.

6.4.1 Multiple Inheritance Rules and Guidelines

Multiple inheritance refers to the ability of a subclass to be formed from more than one superclass. That is, the subclass inherits the operations, attributes, notifications, packages and behaviour from more than one superclass. Multiple inheritance is regarded as an optional facility to be used by object designers when it is appropriate or useful.

Strict inheritance ensures that when a class has inherited the same characteristics from multiple superclasses, then the class is defined as though that characteristic were inherited from only a single superclass. If the characteristic is a conditional package in any of the multiple superclasses but is mandatory in at least one of the multiple superclasses, then it is mandatory in the subclass.

Inheritance from multiple superclasses shall not result in contradictions in the subclass definition. Subclause 10.3.3 of ITU-T Recommendation X.722 [8] GDMO provides descriptions of the kinds of contradictions that can occur and provides clear guidelines for their resolution.

Use multiple inheritance to combine several instantiable managed object classes into a powerful new subclass.

Although multiple inheritance provides a means of gathering collections of characteristics into one managed object class, it must not be used to define subclasses where name bindings are the appropriate approach.

Use multiple inheritance to create a multi-service managed object class unless the services are more appropriately modelled in a conditional package. As a general rule, conditional packages are preferred when a subset of very similar managed objects contains the services while the rest do not for example a resource comes in several models with some models offering more features or a software package that has two or more releases) or when the set of services seems too limited or non-unitary to be defined as a managed object class.

6.4.2 Multiple Inheritance

Multiple inheritance must not be used to model a containment relationship when that relationship can be logically modelled using name bindings and the containment tree. Use multiple inheritance to combine the characteristics of two or more managed object classes under the following conditions only:

- the managed object classes are not logically related via an IS-PART-OF relationship as described in the text above;
- multiple copies of the same characteristic of the managed object class under consideration for multiple inheritance will never be required in any instances of the second managed object; or
- references to the entire managed object that was multiply inherited into another managed object will never be necessary.

6.5 Use of Allomorphism

There are circumstances in which managing systems capable to manage a certain set of managed objects are expected - at least to the extent to the standard features - to manage extended variants of those. In a multi-vendor network, for example, there are equipments from several suppliers adding different proprietary specializations to the standard definitions.

Despite the standardization efforts to help the development of interworking management systems, unless compatibility of the objects they manage are not implemented on a common way, standard systems are still interoperable only to the extent they share the knowledge about specialized managed object classes in their models.

The OSI Management Information Model defined in ITU-T Recommendation X.720 [6] identifies the required techniques to provide interoperability for management systems containing extended or proprietary but compatible managed objects. Different ways though, interoperability may be provided by both managed and managing systems. Interoperability requires from a managing system a certain level of flexibility to ignore any extra information it receives from managed systems. In this approach - which is often termed best-effort management - if, for example, the managing system requests by scoping attribute values of a specialized object not known by the system, it is possible to decide from the contents of attribute allomorphs (which is an attribute of top object class and as such inherited into all objects) which class in the list the system can manage and ignore any attribute which is outside the scope of that particular object class. For that reason it is advisable to always select allomorphs in requests where handling of unknown managed object classes might become necessary.

Managed systems may support interoperability by implementing allomorphism in managed objects.

Allomorphism is the object modelling tool which enables an object to imitate another class containing a subset of the attributes, notifications and operations present in the allomorphic class. For example a signalling link termination point object could be managed as a termination point object. Thus a management system, without knowledge of the new object class can manage that object as if it were its superclass (with multiple inheritance an object may have more than one superclass which it is able to imitate).

The allomorphic behaviour of managed objects as it is described in ITU-T Recommendation X.720 [6] ISO/IEC 10165-1 [14] requires to maintain the allomorphs attribute defined in ITU-T Recommendation X.721 [7] ISO/IEC 10165-2 [15]. In case of specialized objects it is also recommended in M-CREATE response to reply the attribute list as it may ease the managing system to associate the newly created object instance to the right object class later on.

In alignment with recommendations concerning allomorphism, management services defined in ITU-T Recommendation X.710 [16] specify both object class and object instance as parameters to operation requests where allomorphic behaviour may be expected. This way the managed system is informed which subset of the features supported by the selected object instance the managing system is interested in, provided the object maintains allomorphism with the object class specified.

Note that although allomorphism is not limited to inheritance, implementing it along the inheritance tree only is advisable as it helps integratibility.

6.6 Use of Relationships

6.6.1 General Rules and Guidelines

General rules are:

- provide a consistent approach to represent all kinds of relationships and their properties externally from the MO classes. The relationship definition approach should also include cardinality;
- provide a graphical expression of all system relationships;
- name objects from the root (or system equivalent) when no containment relationship needs to be captured. This releases the relationship from undue or misleading significance;
- define naming structure to utilize scoping and filtering;
- when multiple Name bindings are defined to apply on a single MO class, it should be stated clearly in the profile which name binding applies in which circumstances.

There are three ways to specify one of the above relations depending on the complexity of the actual relationship. These are; relation attributes, relation objects, and name binding. Select the modelling method that best represents the underlying relation among the resources being modelled.

Relationships can be modelled using relationship attributes defined in the managed object classes which participate in the relation. Although this method is simple and intuitive, it only captures the semantics of the relation and the managed object instances involved. Specific characteristics of the relationship are not maintained.

Relationships can be modelled with managed object classes. This allows the relationship to be modelled separately from the objects which participate in the relationship and allows characteristics of the relationship to be modelled. This approach is more extensible than the use of relationship attributes.

Name bindings may be used to indicate those containment/existence dependency relationships which are suitable for naming.

Ensure that dual pointers are both changed at the same time, and that consistency is maintained.

Use containment or other relationships when a managed object instance must participate in relationships with multiple instances of the same class (it is not possible to have multiple individual or group attributes of the same type in a managed object class).

Use individual managed object classes and relationships between them to model aspects of structure that must change over time.

If the naming tree cannot represent all of the containment relationships, represent the most static tree structure with naming, and use relationship attributes or objects.

6.6.2 Modelling Relations with Attributes

Relations may be specified within the managed object class using a relationship attribute. This is suitable when the relation is known explicitly at the time of object definition. The attributes may be single or set-valued, depending on whether one or many managed object instances are involved in a particular type of relation.

6.6.3 Modelling Relations with Managed Object Classes

Some relationships may be better specified separately from the classes which participate in the relationship. These include:

- relationships which specify an effect on the behaviour of one or more other managed objects;
- relationships which are expected to become more complex for example by the addition of more related classes and where this complexity is expected to extend the current situation. In this case, a new subclass specification would be required and future users would uses the new subclass specification;
- relationships which are many-to-many in nature. These can only be deduced by aggregating all the one-to-many relationships which are defined by pointing;

- where it is advantageous to keep all the information relating to a particular relationship in a single managed object in order to access or manipulate it more easily;
- where it is useful to define a managed object class to model properties of a relationship which are not properties of the individual managed objects participating in the relationship.

Once a managed object class has been defined, relationships can only be added by specification elsewhere. There are two basic categories of additional relationships as follows:

- A new managed object class is developed which has a new relationship with an existing managed object class. In this case it may be suitable to incorporate the relationship specification into the new managed object class when it is defined.
- A new relationship, or a variation on an old one, is to be modelled between instances of existing classes (it could be instances of a single class). This specification must be captured in a new managed object class.

It is possible to provide indirect relationship pointers by two methods. These allow searching to discover relations, but do not allow direct manipulation of the relation.

- An intermediate object may be defined which holds a list of all the managed objects which are related to a particular managed object. The nature of these relationships is defined in one, or both, of the related managed objects rather than in the intermediate object.
- A set-valued attribute may be defined for a managed object class (potentially for every class) which points to every related managed object instance.

6.6.4 Modelling Relations with Name Bindings

There are two conditions which designers should take into consideration when defining relationships using name bindings. These are:

- a naming relationship actually exists between two instances. This is not necessarily the same as the existence of a containment relationship between two instances since it is possible to have containment relationships without naming relationships;
- the name binding should be unambiguous. It is currently possible to specify multiple name bindings for a given pair of managed object classes, each of which may specify the containment relationship with different semantics.

Managed Object instances are named (unambiguously identified) through the use of a chosen set of containment relationships. These containment relationships form a hierarchy but must also satisfy two other properties:

- each managed object is contained in only one other managed object, however a managed object may contain many other objects;
- a managed object can only exist if its containing (superior) managed object exists.

Any set of containment relationships satisfying these properties may be used for naming.

NOTE 1: The containment relationships used for naming do not reflect physical containment.

NOTE 2: The inheritance hierarchy is completely independent of the naming tree.

The name of a managed object instance is based on the naming tree and is a combination of the name of the superior plus information uniquely identifying the managed object within the scope of its superior managed object.

Naming - ETSI Specific Aspects

Objects reflecting the Network Element Viewpoint (that information required to manage a network element) shall be named through the ETSI Managed Element Object class or a subclass of the Managed Element Object class.

Objects reflecting the Network Level Viewpoint (that information required to manage the physical and logical network) shall be named through the Network Object class or a subclass of the Network Object class.

NOTE 3: Specific naming constraints and the relationships these represent should be described as part of behaviour. The use of subclasses with different naming constraints is an inefficient mechanism to replace that which can easily be achieved as part of behaviour.

6.7 Use of Structuring

This subclause summarizes the modelling principles that deal with reflecting the inherent structure of network and system resources in managed object instances. In this instance, the class hierarchy cannot be used because it no longer exists during execution. Instead the managed object designer must rely on the structuring capabilities that are inherent in the components of the managed object class (single attributes, group attributes, and packages), and the fact that managed object classes can be recursively nested. The subclauses below summarize the guidelines that must be used to reflect resource data structure in the managed object class definitions.

When defining a significant portion of an inheritance hierarchy, it is useful to define a separate object class (using informal syntax and semantics) for every resource that needs to be managed. After defining all objects informally, the specifications can be reviewed for commonality of characteristics. Common aspects can then be "factored out" into superclasses which can then be organized into the inheritance hierarchy.

Begin the process of modelling a set of resources by specifying each aspect that is to be modelled in each resource. Once this is accomplished, apply the following:

- factor out any common aspects (attributes, operations, notifications) and organize them into superclasses;
- identify any common aspects that are better represented as conditional packages instead of superclasses;
- when identifying potential superclasses or conditional packages, use previously defined class hierarchies (for example ISO object definition, etc.) to guide selection. This will ensure that the newly generated hierarchy is consistent with previous work.

When constructing superclasses or conditional packages according to the above guidelines, a key consideration is interoperability. Anticipating and accommodating extensions, via subclassing, to other technologies or vendor implementations of the resource being modelled is very important and ensures the generation of generic superclasses useful to other object designers.

Use inheritance to organize class specifications for reusability, but not to make resource structure visible in managed object instances.

Modelling Managed Properties of a Resource

It is preferable to define each managed property of a resource (represented by an object) as an attribute and operate on this attribute with the defined operations (i.e. GET, REPLACE, ADD, REMOVE, SET TO DEFAULT). Hence correlation of management operations on these properties could be represented in a similar way (for example by state table) and the behaviour description will be simplified.

The use of actions (operations acting on a resource) without representing each property as an attribute is not recommended as there is no formal way of describing the effect of the action on the object to provide a conformance statement.

6.8 Use of Optionality

This subclause summarizes the guidelines for the use of optionality as a specification tool when defining managed object classes. Optionality is treated in its most general sense - meaning choice in the management process, conditional presence and ambiguity. Ambiguity is treated since the major drawback of inappropriately used optionality manifests itself in ambiguity of specification.

- Optionality should only exist when there is a clear need for it.
- A causal optionality (the kind where the reason for the choice is not readily apparent to an automated task) should be used only when it does not affect determination of what activity should occur next. This is either because the optional information is not relevant to a choice in management or because the optional information will always be processed by a human.
- Do not use conditional packages to model IS-A relations in a class hierarchy. IS-A relations are modelled using strict inheritance. Exceptions to this rule should be carefully justified since it is tempting to use conditional inheritance to bypass the strict inheritance rule.

If null, or other special value attributes are used, it should be made clear, in the behaviour, the meaning of the value. The use of special value attributes having a special meaning is discouraged.

6.9 Behaviour Specification

This subclause summarizes modelling guidelines for behavioural specifications in plain English (SDL & Z behaviours will be different). These guidelines are in addition to those in ITU-T Recommendation X.720 [6].

Behaviour descriptions should be structured as follows:

- general behaviour;
- relationship behaviour;
- create behaviour;
- delete behaviour).

To supply additional behaviour only to an object, create a subclass and add a new mandatory package containing just behaviour and named in such a way to indicate that it refers to the object class (for example objectClassName Package).

The following guidelines should be also followed when describing behaviour:

- There should be a clear distinction between "comments" and the "actual behaviour" of the managed object.
- Behaviour definitions should be rigorous, and expressed in informal text with reference to semi-formal methods where required.
- Where a behaviour references attributes, actions, notifications or parameters their identity should be unambiguously specified.
- When defining behaviour using informal text, the language used should avoid an over descriptive style; the definition should be "matter of fact". Generalizations should be avoided - words and phrases like: "normal", "in general", "made up of", "often", "many times", "sometimes", "etc.".
- Behaviour should not repeat what is already adequately specified by the syntax used (GDMO or ASN.1). Rather than clarify the meaning, text can often contain a mistake which is simply confusing and misleading. For example, it is very easy to make an error in transcribing a list of enumerations.
- Behaviour definitions are as long as they need to be. Specifiers should continue to define behaviour until they are satisfied that it is complete. It is not unreasonable to expect the behaviour definition to be significantly longer than the associated GDMO.
- Use of diagrams as comments to explain behaviour is strongly recommended.
- Particular attention should be paid to defining ACTION behaviours. The specifier must ensure that all failure modes are defined, as well as success, and that any changes in object state are clearly defined.
- Behavioural assertions that apply to several classes should be stated as early as possible in the class hierarchy. This promotes reusability and encapsulation.
- A behavioural frame of reference (i.e. in mathematical terms, a "set of states") of a managed object class is defined by the values of all its attributes and by all the values of relevant attributes of any managed objects that are related to the managed object instance at the time defined by the behavioural frame of reference.

6.9.1 Guidelines and Principles for Class Invariants

Instances of every instantiable class can be created, either by a CREATE operation, or by a local method which initializes an instance of the class, including all its attributes.

If there are no attribute defaults or initialized values, then the missing values must be supplied at the time of creation. This may be done by an explicit CREATE operation or by a local mechanism.

6.9.2 Behavioural Preconditions and Postconditions

Use pre and postcondition clauses in GDMO, these will reflect the dynamic pre and postconditions.

Pre and postconditions may use Class Invariants. The Class Invariant is inherited into all subclasses of the class where it is defined. The class invariant must be preserved without change in all descendant subclasses of the class where it was first defined. No subclass may redefine an assertion that is part of a superclass's Class Invariant. Hence it is important that when constructing a Class Invariant for a managed object class, the impact it will have on any subclasses that will or might be derived from that class is very carefully considered.

Behaviour descriptions should not include any general statements on instanciability (for example "this object class is not instanciable") since this is inherited by all subclasses. Restrictions should be specific (for example naming the object class) or contained in the introductory text to the object.

Preconditions

The precondition for an operation or notification is always stated in the form of requirements that must be satisfied before the operation can be performed.

The precondition for an operation or notification may be modified in the subclass to reflect the modification to the semantics of the operation or notification. However the modified precondition must be no stronger than the original precondition in the superclass.

Postconditions

If an attribute's value changes after an operation or notification, the change from the old value to the new value is explicitly described in the behavioural postcondition. If an attribute is not mentioned in the postcondition, this is a tacit guarantee that the attribute's value has not changed.

The postcondition for an operation or notification may be modified in the subclass to reflect the modification to the semantics of the operation or notification. However the modified postcondition must be no weaker than the original postcondition in the superclass.

6.10 Use of Packages

Packages consisting of a combination of behaviour definitions, attributes, attribute groups, operations, notifications and parameters may be defined for subsequent insertion into a managed object under the Characterized By or Conditional Package constructs.

All external packages are registered in an attribute inherited from Top, therefore all packages likely to be reused need to be registered.

6.10.1 Mandatory Packages

Each object shall contain at least one mandatory (Characterized By) package. The use of multiple packages is an equivalent alternative to multiple inheritance. The specification of mandatory packages is encouraged to assist in profiling.

6.10.2 Conditional Packages (X.720 subclause 5.1.5)

The PRESENT IF clause should include a meaningful condition preferably (NOT "if an instance supports it") unless it is intended to indicate optionality. For example the condition could be based on support of a particular function or relationship.

An example of a "good" conditionality statement is the crossConnectionPointerPackage from the ITU-T Recommendation M.3100 [1]:1992 Termination Point Object Class:

- crossConnectionPointerPackage PRESENT IF;
- "the termination point can be flexibly assigned, (i.e. cross connected)".

A profile should always be provided for a particular implementation, this profile should state unambiguously which conditional packages are used, and which are excluded.

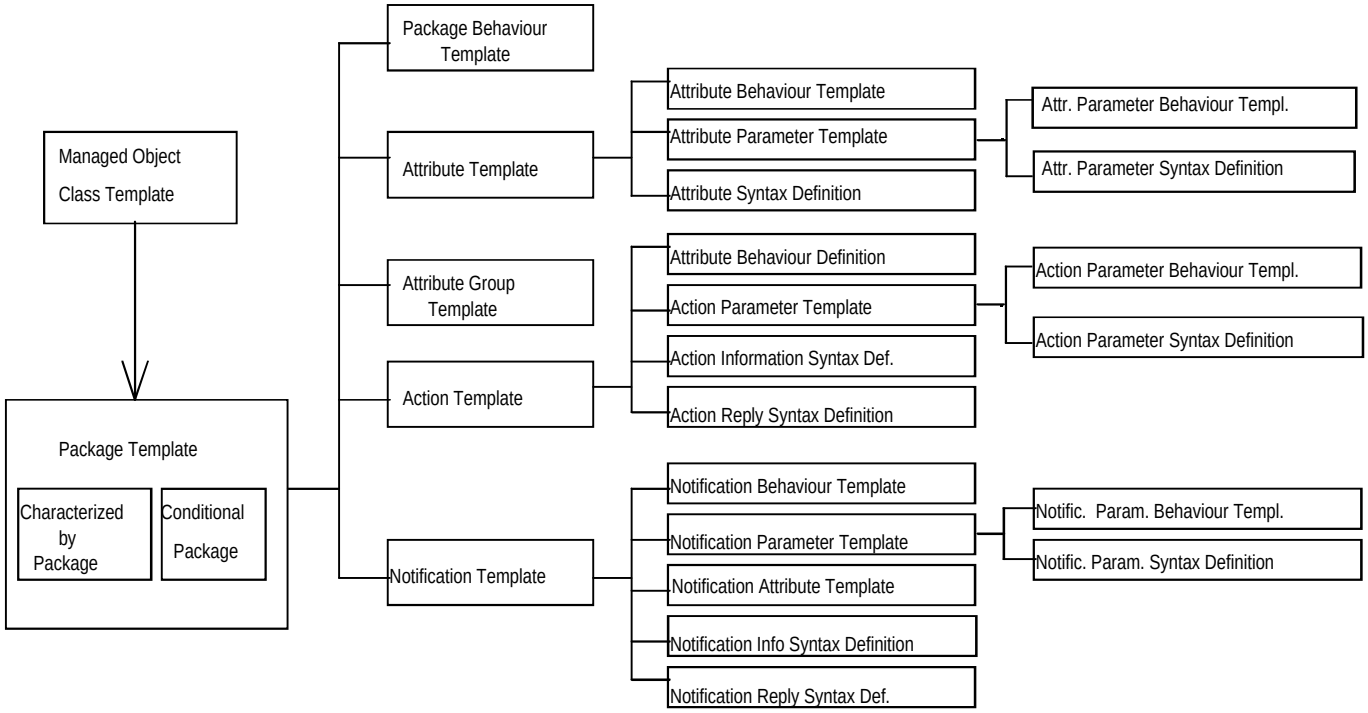


Figure 6: Template Overview for the Definition of Managed Object Classes

Bibliography

The following material, though not specifically referenced in the body of the present document (or not publicly available), gives supporting information.

ITU-T Recommendations

ITU-T Recommendation X.732: "Information technology – Open Systems Interconnection – Systems Management: Attributes for representing relationships".

ITU-T Recommendation X.750: "Information technology – Open Systems Interconnection – Systems Management Management knowledge management function".

EWOS Technical Reports

EWOS: "Guidelines for Managed Object Taxonomy and Profiles".

EWOS: "Framework for Conformance and testing of OSI Management Profiles".

EWOS: "Guidelines for Scope and Content of Ensembles".

NMF Technical Reports

NM Forum TR102 (1991): "Modelling Principles for Managed Objects".

ECMA Technical Reports

ECMA: "Guidelines for the Definition of Managed Object Classes used in PTNs".

Technical Books

Rumbaugh et al: "Object-oriented Modeling and Design" Prentice-Hall 1991.

J.M. Spivey: "The Z Notation" Prentice-Hall 1992.

Potter et al: "An Introduction to Formal Specification and Z" Prentice-Hall.

Generic Models

I-ETS 300 293: "Telecommunications Management Network (TMN); Generic managed objects".

I-ETS 300 653: "Telecommunications Management Network (TMN); Generic managed object class library for the network level view".

ETR 088: "Network Aspects (NA); Time/type of day dependant scheduling function support object classes".

Q3 Interface Models

I-ETS 300 291: "Network Aspects (NA); Functional specification of Customer Administration (CA) on the Operations System/Network Element (OS/NE) interface".

I-ETS 300 292: "Network Aspects (NA); Functional specification of call routeing information management on the Operations System/Network Element (OS/NE) interface".

I-ETS 300 637: "Network Aspects (NA); Functional specification of traffic management on the Network Element/Operations System (NE/OS) interface".

I-ETS 300 819: "Telecommunications Management Network (TMN); Functional specification of usage metering information management on the Operations System/Network Element (OS/NE) interface".

Technology Specific Models

ITU-T Recommendation G.773: "Protocol suites for Q interfaces for management of transmission systems".

ETS 300 150: "Transmission and Multiplexing (TM); Protocol suites for Q interfaces for management of transmission systems".

ITU-T Recommendation G.784: "Synchronous digital hierarchy (SDH) management".

ITU-T Recommendation G.831: "Management capabilities of transport networks based on the Synchronous Digital Hierarchy (SDH)".

ETS 300 304: "Transmission and Multiplexing (TM); Synchronous Digital Hierarchy (SDH); SDH information model for the Network Element (NE) view".

ITU-T Recommendation G.774: "Synchronous digital hierarchy (SDH) management information model for the network element view".

ITU-T Recommendation G.774.01: "Synchronous Digital Hierarchy (SDH) performance monitoring for the network element view".

ETS 300 411: "Transmission and Multiplexing (TM); Performance monitoring; Information model for the Network Element (NE) view".

ITU-T Recommendation G.774.02: "Synchronous digital hierarchy (SDH) configuration of the payload structure for the network element view".

ETS 300 412: "Transmission and Multiplexing (TM); Payload configuration; Information model for the Network Element (NE) view".

ITU-T Recommendation G.774.03: "Synchronous digital hierarchy (SDH) management of multiplex-section protection for the network element view".

ITU-T Recommendation G.783: "Characteristics of synchronous digital hierarchy (SDH) equipment functional blocks".

ETS 300 413: "Transmission and Multiplexing (TM); Multiplex section protection; Information model for the Network Element (NE) view".

ITU-T Recommendation G.774.05: "Synchronous Digital Hierarchy (SDH) management of connection supervision functionality (HCS/LCS) for the network element view".

ETS 300 484: "Transmission and Multiplexing (TM); Synchronous Digital Hierarchy (SDH) network information model; Connection supervision function (Higher order Connection Supervision / Lower order Connection Supervision (HCS/LCS)) for the Network Element (NE) view".

ITU-T Recommendation G.774.04: "Synchronous digital hierarchy (SDH) management of the subnetwork connection protection for the network element view".

ETS 300 493: "Transmission and Multiplexing (TM); Synchronous Digital Hierarchy (SDH) information model of the Sub Network Connection Protection (SNCP) for the Network Element (NE) view".

ETS 300 371: "Transmission and Multiplexing (TM); Plesiochronous Digital Hierarchy (PDH) information model for the Network Element (NE) view".

ETR 240: "Transmission and Multiplexing (TM); Optical Access Networks (OANs); Operations and Maintenance (OAM) of OANs".

ETR 062: "Network Aspects (NA); Baseline document on the integration of Intelligent Network (IN) and Telecommunication Management Network (TMN)".

ETR 224: "Intelligent Network (IN); IN Capability Set 2 (CS2); IN intra-domain management requirements for CS2".

I-ETS 300 291: "Network Aspects (NA); Functional specification of Customer Administration (CA) on the Operations System/Network Element (OS/NE) interface".

I-ETS 300 292: "Network Aspects (NA); Functional specification of call routeing information management on the Operations System/Network Element (OS/NE) interface".

I-ETS 300 637: "Network Aspects (NA); Functional specification of traffic management on the Network Element/Operations System (NE/OS) interface".

ETS 300 324-1: "V interfaces at the digital Local Exchange (LE); V5.1 interface for the support of Access Network (AN); Part 1: V5.1 interface specification".

ETS 300 347-1: "V interfaces at the digital Local Exchange (LE); V5.2 interface for the support of Access Network (AN); Part 1: V5.2 interface specification".

ETS 300 376-1: "Signalling Protocols and Switching (SPS); Q3 interface at the Access Network (AN) for configuration management of V5 interfaces and associated user ports; Part 1: Q3 interface specification".

ETS 300 376-2: "Q3 interface at the Access Network (AN) for configuration management of V5 interfaces and associated user ports; Part 2: Managed Object Conformance Statement (MOCS) proforma specification".

ETS 300 377-1: "Q3 interface at the Local Exchange (LE) for configuration management of V5 interfaces and associated customer profiles; Part 1: Q3 interface specification".

ETS 300 377-2: "Q3 interface at the Local Exchange (LE) for configuration management of V5 interfaces and associated customer profiles; Part 2: Managed Objects Conformance Statement (MOCS) proforma specification".

ETS 300 378-1: "Q3 interface at the Access Network (AN) for fault and performance management of V5 interfaces and associated user ports; Part 1: Q3 interface specification".

ETS 300 379-1: "Signalling Protocols and Switching (SPS); Q3 interface at the Local Exchange (LE) for fault and performance management of V5 interfaces and associated customer profiles; Part 1: Q3 interface specification".

ITU-T Recommendation Q.750: "Overview of Signalling System No. 7 management".

ITU-T Recommendation Q.751: "Signalling System No. 7 managed objects".

ITU-T Recommendation Q.752: "Monitoring and measurements for Signalling System No. 7 networks".

ITU-T Recommendation Q.753: "Signalling System No. 7 management functions MRVT, SRVT and CVT and definition of the OMASE-user".

ITU-T Recommendation Q.754: "Signalling System No. 7 management Application Service Element (ASE) definitions".

ITU-T Recommendation Q.755: "Signalling System No. 7 protocol tests".

ITU-T Recommendation M.4100 (1993): "Maintenance of common channel Signalling System No. 7".

ITU-T Recommendation M.4110 (1993): "Inter-Administration agreements on common channel Signalling System No. 7".

ITU-T Recommendation Q.513: "Digital exchange interfaces for operations, administration and maintenance".

ETS 300 469: "Broadband Integrated Services Digital Network (B-ISDN); Asynchronous Transfer Mode (ATM); Management of the network element view [ITU-T Recommendation I.751 (1996)]".

ITU-T Recommendation I.751: "Asynchronous transfer mode management of the network element view".

CCITT Recommendation Q.940 (1989): "ISDN user-network interface protocol for management -General aspects".

ITU-T Recommendation Q.941 (1994): "Digital subscriber Signalling System No. 1 (DSS 1) - ISDN user-network interface protocol profile for management".

ETR 179: "Universal Personal Telecommunication (UPT); UPT management aspects".

ETS 300 612-1 (1996): "Digital cellular telecommunications system (Phase 2); Network Management (NM); Part 1: Objectives and structure of network management (GSM 12.00)".

ETS 300 612-2 (1996): "Digital cellular telecommunications system (Phase 2); Network Management (NM); Part 2: Common aspects of GSM/DCS 1800 network management (GSM 12.01)".

ETS 300 613 (1996): "Digital cellular telecommunications system (Phase 2); Subscriber, Mobile Equipment (ME) and services data administration (GSM 12.02)".

ETS 300 614 (1996): "Digital cellular telecommunications system (Phase 2); Security management (GSM 12.03)".

ETS 300 615 (1996): "Digital cellular telecommunications system (Phase 2); Performance data measurements (GSM 12.04)".

ETS 300 616: "Digital cellular telecommunications system (Phase 2); Event and call data (GSM 12.05 version 4.3.1)".

ETS 300 617: "Digital cellular telecommunications system (Phase 2); GSM network configuration management (GSM 12.06)".

ETS 300 627: "Digital cellular telecommunications system (Phase 2); Subscriber and equipment trace (GSM 12.08 version 4.5.1)".

ETS 300 622: "Digital cellular telecommunications system (Phase 2); Base Station System (BSS) management information (GSM 12.20)".

ETS 300 623: "Digital cellular telecommunications system (Phase 2); Network Management (NM) procedures and messages on the A-bis interface (GSM 12.21)".

ETS 300 624: "Digital cellular telecommunications system (Phase 2); Interworking of GSM Network Management (NM) procedures and messages at the Base Station Controller (BSC) (GSM 12.22)".

ETR 128: "European digital cellular telecommunications system (Phase 2); ETSI Object Identifier tree; Common domain; Mobile domain; Operation and Maintenance (O&M), managed object registration definition (GSM 12.30)".

ETS 300 133-7/A1 (1994): "Paging Systems (PS); Enhanced Radio MEssage System (ERMES); Part 7: Operations and maintenance aspects".

ECMA TR/54: "A Management Framework for PTNs".

ETR 245: "Private Integrated Services Network (PISN) management; Compendium of PISN management services".

ITU-T Recommendation X.160: "Architecture for customer network management service for public data networks".

ITU-T Recommendation X.161: "Definition of customer network management services for public data networks".

ITU-T Recommendation X.162: "Definition of management information for customer network management service for public data networks to be used with the CNMc interface".

ITU-T Recommendation X.163: "Definition of management information for customer network management service for public data networks to be used with the CNMe interface".

History

Document history		
Edition 1	July 1992	Publication as ETR 046
V1.2.1	February 1999	Publication

ISBN 2-7437-2840-X
Dépôt légal : Février 1999