

JT-X722
OSIネットワーク管理
- 管理オブジェクト定義ガイドライン

OSI Network Management
- Guidelines for the Definition of Managed Object

第1版

1994年4月27日制定

社団法人
情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

本書は、(社)情報通信技術委員会が著作権を保有しています。
内容の一部又は全部を(社)情報通信技術委員会の許諾を得ることなく複製、転載、改変、
転用及びネットワーク上での送信、配布を行うことを禁止します。

<参考>

1. 国際標準との関係

- (1) 大規模、マルチベンダー化するネットワークを共通的なサービス/プロトコルで管理するため、OSIネットワーク管理の標準化が進められている。OSIネットワーク管理を行うためのプロトコルCMIS/CMIPで使用される管理情報を規定する国際標準 Rec. X. 72x | ISO/IEC 10165 (Open Systems Interconnection Structure of Management Information) は、以下のパートから構成される。

part1(X.720) : Management Information Model (管理情報モデル)

part2(X.721) : Definitions of Management Information (管理情報定義)

part4(X.722) : Guidelines for the definition of Managed Object

(管理オブジェクト定義ガイドライン)

part5(X.723) : Generic Management Information (汎用管理情報)

part6(X.724) : Requirement and Guidelines for Implementation Conformance

Statement Proformas associated with Management Information

(管理情報に関する実装適合性宣言プロフォーマの要件およびガイドライン)

本標準は、OSIネットワーク管理を行うための管理オブジェクト定義のガイドラインについて記述しており、1992年版ITU-T勧告X. 722を基本標準とし、JIS (日本工業規格) X 5603の開放型システム間相互接続の抽象構文記法1 (ASN. 1)に基づいた日本語表記について記述した。

2. 上記国際勧告に対する追加項目等

2.1 オプション選択項目

なし

2.2 ナショナルマター決定項目

JIS (日本工業規格) X 5603の開放型システム間相互接続の抽象構文記法1 (ASN. 1)に基づいた日本語表記について記述した。

3. 原勧告との章立て構成比較表

前記国際勧告との章立て構成の対比を以下に示す。

J T - X 7 2 2	国際勧告
1章 適用範囲	ITU-T 勧告 X.722 1章
2章 規範参照文献	ITU-T 勧告 X.722 2章
3章 定義	ITU-T 勧告 X.722 3章
4章 記号と略号	ITU-T 勧告 X.722 4章
5章 記法	ITU-T 勧告 X.722 5章
6章 概要	ITU-T 勧告 X.722 6章
7章 全体的な検討事項	ITU-T 勧告 X.722 7章
8章 管理オブジェクト定義の一般的原則	ITU-T 勧告 X.722 8章
9章 管理オブジェクト定義のための 記述ツール	ITU-T 勧告 X.722 9章
資料A ガイドラインの使用例	ITU-T 勧告 X.722 Annex A

4. 改版の履歴

版数	制定日	改版内容
第1版	平成 6年 4月27日	制定

5. 工業所有権

本標準に関わる「工業所有権の実施の権利に係る確認書」の提出状況は、TTCホームページでご覧になれます。

6. その他

(1) 参照している勧告、標準等

- ・ TTC 標準 :
- ・ ITU - T 勧告 : X . 2 0 0、X . 2 0 8
X . 5 0 1、
X . 7 0 0、X . 7 0 1、
X . 7 1 0、X . 7 1 1、
X . 7 2 0、X . 7 2 1、
X . 7 3 2、X . 7 3 3、
- ・ ISO 国際標準 : ISO / IEC 7 4 9 8 - 3、
ISO / IEC 9 5 9 6、
ISO / IEC 9 8 3 4 - 1

(2) その他

参照している勧告、国際標準との内容に差異がある場合は、参照している勧告、国際標準等が優先するものとする。

目 次

1. 適用範囲 (Scope)	1
2. 規範参照文献 (Normative reference)	2
2.1 勧告 国際標準 (共通テキスト) (Identical ITU-T Recommendations International Standards)	2
2.2 ITU-T 勧告 国際標準 (同一内容テキスト) (Paired ITU-T Recommendations International Standards)	3
2.3 追加参照文献 (Additional references)	3
3. 定義 (Definitions)	4
3.1 基本参照モデル定義 (Basic Reference Model definitions)	4
3.2 ネーミングとアドレッシングの定義 (Naming and addressing definitions)	4
3.3 管理構造の定義 (Management Framework definitions)	4
3.4 システム管理概要の定義 (Systems Management Overview definitions)	4
3.5 管理情報モデル定義 (Management Information Model definitions)	5
3.6 CMIS 定義 (CMIS definitions)	5
3.7 ASN.1 定義 (ASN.1 definitions)	6
3.8 追加定義 (Additional definitions)	6
3.8.1 管理オブジェクトクラス定義 (Managed object class definition)	6
3.8.2 テンプレート (Template)	6
3.8.3 ディレクトリオブジェクトクラス (Directory object class)	6
4. 記号と略号 (Symbols and abbreviation)	6
5. 記法 (Conventions)	7
6. 概要 (Overview)	7
7. 全体的な検討事項(Global issues)	8
7.1 関係の整合性(integrity)	8
7.2 継承特性 (Inherited characteristics)	8
7.3 オプション性 (Optionality)	8
7.4 登録 (Registration)	9
7.4.1 システム管理概要に対するオブジェクト識別子の割り当て (Object identifier allocation for Systems Management Overview)	10
7.4.2 CMIP に対するオブジェクト識別子の割り当て (Object identifier allocation for CMIP)	11
7.4.3 機能標準用オブジェクト識別子割り当て (Object identifier allocation for function standards)	12
7.4.4 SMI 標準のオブジェクト識別子割り当て (Object identifier allocation for SMI standards)	13
7.4.5 実クラスに対するオブジェクト識別子の割り当て (Object identifier allocation for actual class)	13
7.5 適合性 (Conformance)	14
7.6 管理オブジェクト定義の複雑さ (Complexity of managed object definition)	14
7.7 管理オブジェクトの生成と消去 (Managed object creation and deletion)	14
7.7.1 初期値管理オブジェクト(Initial Value Managed Objects)	14
7.7.2 初期属性値のソース (Sources of initial attribute values)	14
8. 管理オブジェクト定義の一般的原則 (General principles for managed object definition)	15
8.1 共通性 (Commonality)	15

8.2	管理すべきもの (What to manage)	15
8.3	構造化 (Structuring)	15
8.4	管理オブジェクト (Managed objects)	16
8.4.1	スーパークラスのインスタンス生成 (Instantiation of superclasses)	16
8.4.2	制限を与えないスーパークラス (Unrestricted superclasses)	17
8.5	属性 (Attributes)	17
8.5.1	属性値集合 (Attribute value sets)	17
8.5.2	属性型 (Attribute types)	18
8.6	属性値の関係 (Attribute value relationships)	18
8.7	S A P のモデル化 (Modeling of SAPs)	18
8.8	統計 (Statistics)	19
8.8.1	一貫性 (Consistency)	19
8.8.2	P D U カウンタ (PDU counters)	19
8.8.3	重複 (Overlaps)	19
8.8.4	非リセットブルカウンタ (Non-resettable counters)	19
8.8.5	事象カウンタ (Event counters)	20
8.9	カウンタ (Counters)	20
8.10	タイマ (Timers)	20
8.11	属性の更新 (Updating of attributes)	20
8.12	属性の精度 (Precision of attributes)	20
8.13	管理オブジェクトの識別 (Managed object identification)	21
8.14	通知 (Notifications)	21
8.14.1	サービスの拒否 (Denial of service)	21
8.14.2	情報の保持 (Information conservation)	21
8.15	操作の使用法 (Use of operations)	22
9.	管理オブジェクト定義のための記述ツール (Notation tools for managed object definition)	23
9.1	記述ツールの概要 (Overview of notational tools)	23
9.2	テンプレート定義で用いられる規約 (Conventions used in template definitions)	23
9.3	管理オブジェクトクラステンプレート (Managed object class template)	28
9.3.1	概要 (Overview)	28
9.3.1.1	継承 (Inheritance)	28
9.3.1.2	必須パッケージ (Mandatory packages)	28
9.3.1.3	条件付きパッケージ (Conditional packages)	28
9.3.1.4	クラスの名前付け (Class naming)	28
9.3.2	テンプレート構造 (Template structure)	28
9.3.3	サポートする定義 (Supporting definition)	29
9.4	パッケージテンプレート (Package template)	31
9.4.1	概要 (Overview)	31
9.4.1.1	振舞い (Behavior)	32
9.4.1.2	包含される属性 (Contained attribute)	32
9.4.1.3	操作と通知 (Operations and Notifications)	32
9.4.2	テンプレート構造 (Template structure)	33
9.4.3	サポートする定義 (Supporting definitions)	34

9.5	パラメータテンプレート (Parameter template)	36
9.5.1	概要 (Overview)	36
9.5.1.1	コンテキスト定義 (Context definition)	36
9.5.1.1.1	アクション情報/応答、イベント情報/応答、特定エラー (Action Info/Reply, Event Info/Reply, Specific Error)	36
9.5.1.1.2	コンテキストキーワード (Context Keyword)	36
9.5.1.1.3	他のテンプレートにおける使用 (Usage in other template)	37
9.5.1.2	シンタックス定義 (Syntax definition)	37
9.5.1.3	属性参照 (Attribute reference)	37
9.5.1.4	振舞い (Behavior)	37
9.5.2	テンプレート構造 (Template structure)	38
9.5.3	サポートする定義 (Supporting definitions)	38
9.6	ネームバインディングテンプレート (Name binding template)	39
9.6.1	概要 (Overview)	39
9.6.2	テンプレート構造 (Template structure)	40
9.6.3	サポートする定義 (Supporting definitions)	40
9.7	属性テンプレート (Attribute template)	42
9.7.1	概要 (Overview)	42
9.7.1.1	導出 (Derivation)	42
9.7.1.2	属性シンタックス (Attribute syntax)	42
9.7.1.3	値の照合 (Value matching)	42
9.7.1.4	振舞い (Behavior)	43
9.7.1.5	属性識別子 (Attribute identifier)	43
9.7.1.6	パラメータ (Parameters)	43
9.7.2	テンプレート構造 (Template structure)	43
9.7.3	サポートする定義 (Supporting definitions)	43
9.8	属性グループテンプレート (Attribute group template)	45
9.8.1	概要 (Overview)	45
9.8.2	テンプレート構造 (Template structure)	45
9.8.3	サポートする定義 (Supporting definitions)	45
9.9	振舞いテンプレート (Behavior template)	47
9.9.1	概要 (Overview)	47
9.9.2	テンプレート構造 (Template structure)	47
9.9.3	サポートする定義 (Supporting definitions)	47
9.10	アクションテンプレート (Action template)	47
9.10.1	概要 (Overview)	47
9.10.1.1	振舞い (Behavior)	47
9.10.1.2	操作モード (Mode of operation)	47
9.10.1.3	抽象構文 (Abstract syntax)	48
9.10.1.4	アクション識別子 (Action identifiers)	48
9.10.1.5	パラメータ (Parameters)	48
9.10.2	テンプレート構造 (Template structure)	48
9.10.3	サポートする定義 (Supporting definitions)	48

9.11 通知テンプレート (Notification template)	49
9.11.1 概要 (Overview)	49
9.11.1.1 振舞い (Behavior)	49
9.11.1.2 抽象構文 (Abstract syntax)	49
9.11.1.3 通知の名前付け (Notification naming)	50
9.11.1.4 パラメータ (Parameters)	50
9.11.2 テンプレート構造 (Template structure)	50
9.11.3 サポートする定義 (Supporting definitions)	50

付録 A

ガイドラインの使用例

A. 1 管理オブジェクトクラス定義 (Managed object class definition)	52
A. 2 ネームバインディング定義 (Name binding definition)	52
A. 3 パラメータ定義 (Parameter definition)	53
A. 4 パッケージ定義 (Package definition)	54
A. 5 属性定義 (Attribute definition)	55
A. 6 属性グループ定義 (Attribute group definition)	55
A. 7 アクション定義 (Action definition)	55
A. 8 通知定義 (Notification definition)	56
A. 9 振舞い定義 (Behavior definition)	57
A. 10 A S N. 1 モジュール (ASN.1 modules)	59

1. 適用範囲 (Scope)

この仕様書は、管理オブジェクトの定義を含む勧告／国際標準の開発者に次のような手引きを提供する。

- a) 管理オブジェクト定義間の一貫性を促進する。
- b) OSI 管理の勧告／標準と整合する方法での定義の開発を保証する。
- c) 有用で共通なドキュメントレイアウト、手順、定義を示すことにより他の作業グループでの重複した努力を減らす。

この目的のために、この仕様書は次のような点を規定している。

- a) 関連する OSI 管理の勧告／国際標準と管理オブジェクトクラスの定義との関係、及びこれらの勧告／国際標準が管理オブジェクトクラスの定義に際しどのように使われるべきかについて
- b) 管理オブジェクトクラスとそれらの属性、通知、アクション、振舞いの定義のために使われる適切な手法について。これらの手法は以下のものを含む。
 - 1) 定義の中でふれておくべき側面の要約
 - 2) 定義の中で使うように推奨されている表記用ツール
 - 3) 定義が従うべき一貫性の指針
- c) 管理プロトコルに対する管理オブジェクトクラス定義の関係と、プロトコル関連の定義からの要求条件
- d) 管理オブジェクトクラス定義のための勧告ドキュメント構成

この仕様書は、次のものを定義するあらゆる勧告／国際標準の開発に適用可能である。

- a) OSI 管理プロトコルによって転送あるいは操作される管理情報
- b) その情報が関連する管理オブジェクト

この仕様書は以下の点については規定及び言及しない。

- a) 機能面から見た管理オブジェクトクラス定義の開発に関する制約、管理オブジェクトクラス定義が関係する勧告／国際標準に関する制約、あるいはある特定の管理環境下での管理オブジェクトクラス定義の使用法に関する制約
- b) リソースを定義するための指針；この資料はリソースの管理ビューを提供する管理オブジェクトの定義を行うための指針のみを与えている。

2. 規範参照文献 (Normative reference)

以下の ITU-T 勧告 | 国際標準には、本文中で参照することにより、この仕様書の規定を構成する規定が含まれている。以下の版数は本仕様書の出版時に有効なものである。全ての勧告／標準は改版されうるものであり、この仕様書の合意関係者は、以下にリストアップされた最新版の勧告 | 標準が適用可能かどうか調査することが望ましい。IEC と ISO のメンバは、現在有効な国際標準の登録簿を保持している。電気通信標準化局は現在有効な ITU-T 勧告の一覧を保持している。

2.1 勧告 | 国際標準 (共通テキスト)

(Identical ITU-T Recommendations | International Standards)

ITU-T Recommendation X.701 | ISO/IEC 10040, Information technology - Open System Interconnection - Systems Management Overview.

ITU-T Recommendation X.720 | ISO/IEC DIS 10165-1, Information technology - Open System Interconnection - Structure of Management Information - Part 1: Management Information Model.

ITU-T Recommendation X.721 | ISO/IEC DIS 10165-2, Information technology - Open System Interconnection - Structure of Management Information - Part 2: Definition of Management Information.

ITU-T Recommendation X.732 | ISO/IEC DIS 10164-3, Information technology - Open System Interconnection - Systems Management - Part 3: Attributes for Representing Relationships.

ITU-T Recommendation X.733 | ISO/IEC DIS 10164-4, Information technology - Open System Interconnection - Systems Management - Part 4: Alarm Reporting Function.

2.2 ITU-T 勧告 | 国際標準 (同一内容テキスト)

(Paired ITU-T Recommendations | International Standards)

ITU-T Recommendation X.200:1988, Reference Model of Open Systems Interconnection for ITU-T Applications.

ISO 7498:1984, Information processing systems - Open Systems Interconnection - Basic Reference Model.

ITU-T Recommendation X.208:1988, Specification of abstract syntax notation one (ASN.1).

ISO/IEC 8824:1990, Information technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One(ASN.1).

ITU-T Recommendation X.501:1988, The Directory Models for ITU-T Applications.

ISO/IEC 9594-2:1988, Information technology - Open Systems Interconnection - The Directory - Part 2: Models

ITU-T Recommendation X.700, Management Framework Definition for Open Systems Interconnection(OSI) for ITU-T Applications.

ISO 7498-4:1989, Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 4: Management Framework.

ITU-T Recommendation X.710:1991, Common Management Information Service Definition for ITU-T Applications.

ISO/IEC 9595:1991, Information technology - Open Systems Interconnection - Common management information service definition.

ITU-T Recommendation X.711:1991, Common Management Information Protocol Specification for ITU-T Applications.

ISO/IEC 9596-1:1991, Information technology - Open Systems Interconnection - Common management information protocol - Part 1: Specification.

2.3 追加参照文献 (Additional references)

ISO 7498-3:1989, Information technology - Open Systems Interconnection - Basic Reference Model - Part 3: Naming and Addressing.

ISO/IEC 9596:1990, Information technology - Open Systems Interconnection - Common management information protocol Specification.

ISO/IEC 9834-1, Procedures for the operation of OSI registration authorities - Part 1: General procedures.

3. 定義 (Definitions)

この仕様書のために、次の定義を適用する。

3.1 基本参照モデル定義 (Basic Reference Model definitions)

この仕様書は Rec. X.200 | ISO 7498 の中で定義された次の用語を利用する。

- a) (N)-コネクション((N)-connection)
- b) (N)-エンティティ((N)-entity)
- c) (N)-レイヤ((N)-layer)
- d) (N)-サービスアクセスポイント((N)-service-access-point)
- e) 開放型システム(open system)
- f) システム管理(systems management)

3.2 ネーミングとアドレッシングの定義 (Naming and addressing definitions)

この仕様書は ISO 7498-3 の中で定義された次の用語を利用する。

- a) (N)-セクタ((N)-selector)

3.3 管理構造の定義 (Management Framework definitions)

この仕様書は ISO 7498-4 の中で定義された次の用語を利用する。

- a) 管理オブジェクト(managed object)
- b) (N)-レイヤ操作((N)-layer operation)

3.4 システム管理概要の定義 (Systems Management Overview definitions)

この仕様書は Rec. X.701 | ISO/IEC 10040 の中で定義された次の用語を利用する。

- a) エージェント(agent)
- b) 一般定義(generic definitions)
- c) 管理オブジェクトクラス(managed object class)
- d) 管理情報(management information)
- e) マネージャ(manager)
- f) (N)-レイヤ管理プロトコル((N)-layer management protocol)
- g) 通知(notification)
- h) 通知型(notification type)
- i) (システム管理) 操作((systems management) operation)
- j) システム管理プロトコル(systems management protocol)

3.5 管理情報モデル定義 (Management Information Model definitions)

この仕様書は Rec. X.720 | ISO/IEC 10165-1 の中で定義された次の用語を利用する。

- a) アクション(action)
- b) 実クラス(actual class)
- c) 属性グループ(attribute group)
- d) 属性識別子(attribute identifier)
- e) 属性型(attribute type)
- f) 属性値集合(attribute value set)
- g) 振舞い(behavior)
- h) 特性(characteristic)
- i) 条件付きパッケージ(conditional package)
- j) 包含(containment)
- k) 継承(inheritance)
- l) 継承階層(inheritance hierarchy)
- m) 初期値管理オブジェクト(initial value managed object)
- n) インスタンス化(instantiation)
- o) 必須パッケージ(mandatory package)
- p) 多重継承(multiple inheritance)
- q) ネームバインディング(name binding)
- r) パッケージ(package)
- s) パラメータ(parameter)
- t) 許容値集合(permitted value set)
- u) 相対識別名(relative distinguished name)
- v) 要求値集合(required value set)
- w) 特殊化(specialization)
- x) サブクラス(subclass)
- y) 下位オブジェクト(subordinate object)
- z) スーパークラス(super class)
- aa) 上位オブジェクト(superior object)

3.6 CMIS 定義 (CMIS definitions)

この仕様書は Rec. X.710 | ISO/IEC 9595 の中で定義された次の用語を利用する。

- a) 属性(attribute)
- b) 共通管理情報サービス(Common Management Information Services)

3.7 ASN.1 定義 (ASN.1 definitions)

この仕様書は Rec. X.208 | ISO/IEC 8824 の中で定義された次の用語を利用する。

- a) オブジェクト識別子(object identifier)
- b) 順序型(sequence type)
- c) 単一型順序型(sequence-of type)
- d) 集合型(set type)
- e) 単一型集合型(set-of type)
- f) 部分型(subtype)
- g) 型(type)
- h) 型参照名(type reference name)
- i) 値参照名(value reference name)

3.8 追加定義 (Additional definitions)

3.8.1 管理オブジェクトクラス定義 (Managed object class definition)

ひとつの管理オブジェクトクラス名が割り当てられる属性/操作/通知及び振舞い定義の集りは、1つの管理オブジェクトクラステンプレートと1つ以上のその他のテンプレートを使って文書化される。これらのテンプレートは本仕様書の中で定義される型を持ち、直接的または間接的に管理オブジェクトクラステンプレートによって参照される。管理オブジェクトクラスの定義は管理オブジェクトクラスのスーパークラスから継承された定義の全ての要素とそのスーパークラスを特殊化した定義の全ての要素を含む(多重継承有り)。

3.8.2 テンプレート (Template)

テンプレートとは、ネームバインディング、管理オブジェクトクラス定義及びその要素(パッケージ、パラメータ、属性、属性グループ、振舞い定義、アクション、通知)の文書化のための標準形式である。

3.8.3 ディレクトリオブジェクトクラス (Directory object class)

Rec. X.501 | ISO/IEC 9594-2 の中で定義されているオブジェクトクラス。

4. 記号と略号 (Symbols and abbreviation)

ASN.1	Abstract Syntax Notation One (抽象構文記法1)
CMIP	Common Management Information Protocol (共通管理情報プロトコル)
CMIS	Common Management Information Service (共通管理情報サービス)
DMI	Definition of Management Information (管理情報定義)
IVMO	Initial Value Managed Object (初期値管理オブジェクト)
MOCS	Managed Object Conformance Statement (管理オブジェクト適合性宣言)
(N)-SAP	(N)-service-access-point ((N)-サービスアクセスポイント)
PDU	Protocol Data Unit (プロトコルデータユニット)
SAP	Service Access Point (サービスアクセスポイント)
SDU	Service Data Unit (サービスデータユニット)
SMI	Structure of Management Information (管理情報構造)
RDN	Relative Distinguished Name (相対識別名)

5. 記法 (Conventions)

本仕様書では、外部で定義された記法はなにも使用していない。

6. 概要 (Overview)

OSI 管理の標準化は、必然的に多数の標準化組織と協調した作業を行なう。ITU-T SGVII と ISO/IEC JTC1 SC21/WG4 は、OSI 管理のアーキテクチャとシステム管理のために使われるサービス、プロトコル、機能及び管理情報構造を記述した勧告と国際標準の開発に共同責任がある。ITU-T、ISO/IEC JTC1 SC21、ISO/IEC JTC1 SC6 等のその他のワーキンググループは、OSI 基本参照モデルの個々のレイヤの管理形態を記述した勧告と国際標準の開発に責任があり、これらは(N)-レイヤ管理プロトコル、(N)-レイヤ操作の管理形態及びレイヤ操作の側面からの「管理ビュー」を与え、システム管理に対して可視となる。

本仕様書は、管理オブジェクトクラス定義の開発者に対して、ISO/IEC と ITU-T との共同によって開発された OSI 管理の標準と矛盾が無い、完全な管理オブジェクトクラスを定義するために要求される情報と文書化のためのツールを提供する。これらのガイドラインを使用する副次的効果は、どこで開発が行なわれるかに関係なく、これらの勧告／標準の文書化に共通なアプローチがとられるようになることである。

Rec.X.700 | ISO/IEC 7498-4 と Rec.X.701 | ISO/IEC10040 は、OSI 管理のアーキテクチャを定義し、OSI 管理の標準の内容と標準間の関係を記述している。

Rec.X.720 | ISO/IEC 10165-1 は、管理オブジェクトの点から管理情報のモデルを記述している。

Rec.X.721 | ISO/IEC 10165-2(DMI)は、汎用管理オブジェクトクラスと特性を記述している。管理情報の適切な定義は DMI にあるので、同様な構造を持つ新しい情報要素の定義に先立ってこの定義を参照することを勧める。

Rec.X.208 | ISO/IEC 8824 は、システム管理プロトコルで転送される、管理オブジェクトの特性に関するデータ要素の抽象構文表現で用いられる記法を定義している。

7. 全体的な検討事項(Global issues)

7.1 関係の整合性(integrity)

管理オブジェクトクラスを定義するとき、それらのクラスのインスタンスに適用される一貫性の要求条件が有る状況を考慮することが重要である。例えば、ある管理オブジェクトの振舞いが自分の状態ばかりではなく、システム中の他の管理オブジェクトの状態にも依存するルールによって制約されるという状況が上げられる。そのようないかなる制約もそれが関係する管理オブジェクトクラスに関する振舞いとして表現されているべきである。

管理オブジェクトクラスのインスタンス生成に関する定義で、明白に一貫性をもったルールを記述すべき特徴的なケースとして、消去操作の定義がある。この操作に対しては、そのような一貫性のルールがその管理オブジェクトクラスに関するネームバインディングで記述される。消去操作による影響はどんな環境で消去が許されているか、消去の結果がどのようなようになるか、が明白となるような方法で定義されていないなければならない。特にネームバインディングでは以下のことを明確にしなければならない。

- ・そのクラスに包含される管理オブジェクトがまだ存在する時に、クラスのインスタンスの消去が許されているかどうか
- ・消去される管理オブジェクトと他の管理オブジェクトの間に他の関係（非包含関係、例えば、Rec.X.732 | ISO/IEC 10164-3 で定義されている関係属性が存在するためにできる関係）が存在している場合どんなルールを適用するか

消去について適用される一貫性のルールは消去操作が矛盾した関係を起こすことがないようにしなくてはならない。これらの一貫性のルールはネームバインディングの一部として指定されるため、ある管理オブジェクトの消去到適用するルールはその管理オブジェクトがインスタンス生成された時点で有効になる。

7.2 継承特性 (Inherited characteristics)

継承のプロセスはある管理オブジェクトクラスのスーパークラス（複数）の全ての特性をその管理オブジェクトクラスの定義の中に含むという結果をもたらす。この規則は再帰的に適用され、top として知られている、継承階層の頂点で終了する。それ故、ある与えられた管理オブジェクトクラスは top の定義部の全ての特性を含み、更にその管理オブジェクトクラスの継承階層部分を形成する top からの全てのサブクラスの定義プロセスで加えられる全ての特性を含んでいる。

7.3 オプション性 (Optionality)

一般に管理オブジェクトクラス定義の中のオプションの設定（訳注：条件付きパッケージの指定）は、相互運用がオプション数の増加によりだんだん難しくなるという理由で勧められない。Rec.X.720 | ISO/IEC 10165-1 で述べられているように、管理オブジェクトクラスの定義は、特定の条件が適用された場合に管理オブジェクトクラスのインスタンスに現れるような、条件付きパッケージを含むことがある。これらのパッケージに適用される条件は、管理オブジェクトクラスが対応するリソースの標準化された特性、あるいは管理システムによってサポートされるオプションな（訳注：SMFの）管理機能と関連すべきであることを意図している。

7.4 登録 (Registration)

管理オブジェクトクラスを定義する過程で、オブジェクト識別子として知られるグローバルにユニークな識別子を管理オブジェクトクラスの様々な側面、例えば、管理オブジェクトクラス名、属性型などに割当てることが要求される。これらの識別子の値は管理プロトコルで使用され、管理オブジェクトとそれに関係する属性、操作、通知の側面をユニークに識別する。それ故、関係する標準化団体 (standards body) や標準化組織 (standards organization) がオブジェクト識別子を発行することが出来る適切な登録機構を明らかにし、または確立することが管理オブジェクトクラス定義の開発に先駆けて必要なことである。Rec.X.208 | ISO/IEC 8824 はオブジェクト識別子の構造と最初の弧 (arc) の値を規定する。登録機構と登録権限者の確立に関する更に多くの情報は ISO/IEC 9834-1 に示されている。

管理情報の項目にオブジェクト識別子の値が一度割当てられたなら、その項目の定義のいかなる改版によっても情報の意味を変えるべきではない。このことは実際には、登録された管理情報の定義に対する編集上の変更は許されるが、その定義がプロトコル上可視的な変更はされるべきではないことを意味する。

システム管理の勧告/標準で登録される全てのオブジェクト識別子の値は弧

{joint-iso-ccitt ms(9)}

の下に割当てられる。

{joint-iso-ccitt ms(9)} の下の弧の割当はこの仕様書で定義される。{joint-iso-ccitt ms(9)} の下の弧はシステム管理の標準毎に割当てられる。以下の弧が割り当てられている。

弧	標準
--	----
smo(0)	システム管理概要、Rec.X.701 ISO/IEC 10040
cmip(1)	共通管理情報プロトコル、Rec.X.711 ISO/IEC 9596-1
function(2)	システム管理機能、Rec.X.7NN ISO/IEC 10164-X、ここで X は ISO/IEC 番号体系での標準のパート番号であり、X.7NN は対応する ITU-T の勧告番号である。
smi(3)	管理情報構造、Rec.X.72N ISO/ICE 10165-X、ここで X は ISO/IEC 番号体系での標準のパート番号であり、X.72N は対応する ITU-T の勧告番号である。

このレベル以下の弧の割当ては 7.4.1-7.4.5 で定義されている。既存のあるいは将来現れるシステム管理の標準によって更に必要となる弧は、要求に応じて、この仕様書の改版 (amendments) によって割り当てられる予定である。

<注> 本節と本節の副節で記述されているオブジェクト識別子の値の割当て方式は ISO/IEC JTC1 SC21/WG4 と ITU-T SGVII とによって共同開発されたシステム管理標準におけるオブジェクト識別子の値の割当てにのみ適用される。管理の標準を開発する過程でそのオブジェクト識別子の値を割当てて必要とする、他の標準化組織や機関は適当な登録権限者のもとで彼ら自身の割当て方式を確立する必要がある。システム管理の標準化活動で採用された構造は、適切な割当て方式がどのように確立されるかの有用な見本として役立つ。ただし、最終的な割当て方式の選択は各関係機関の責任である。オブジェクト識別子の値の可読性を上げるため、Rec.X.208 | ISO/IEC 8824 で定義されているような、オブジェクト識別子の値の表現のためのに名前と数値の形式を、可能な限り使用することが推奨されている。

7.4.1 システム管理概要に対するオブジェクト識別子の割り当て

(Object identifier allocation for Systems Management Overview)

<注>システム管理概要はこれらの弧の割り当てに対して責任がある。それらの弧は情報だけのために含まれている。

{joint-iso-ccitt ms(9) smo(0)}の下に、アプリケーション・コンテキストと抽象構文、ASN.1 モジュール識別子の登録に対して、次のように弧が割り当てられる。

弧	目的
--	----
applicationContext(0)	アプリケーション・コンテキスト識別子の割り当て
negotiationAbstractSyntax(1)	抽象構文のネゴシエーションのためのバージョン識別子の割り当て
asn1Modules(2)	ASN.1 モジュールの識別子の割り当て

{joint-iso-ccitt ms(9) smo(0) applicationContext(0)}の下に更に弧が割り当てられている。これらは Rec.X.701 | ISO/IEC 10040 で仕様化されているように、特定のアプリケーション・コンテキスト識別子の登録のために、次に示すように割り当てられている。

弧	目的
--	----
systems-management(2)	システム管理アプリケーションコンテキストの識別子

{joint-iso-ccitt ms(9) smo(0) negotiationAbstractSyntax(1)}の下に更に弧が割り当てられている。これらは Rec.X.701 | ISO/IEC 10040 で仕様化されているように、特定のバージョンの抽象構文のネゴシエーションの登録に対して、次に示すように割り当てられている。

弧	目的
--	----
version1(1)	抽象構文のネゴシエーションのバージョン1を識別する

{joint-iso-ccitt ms(9) smo(0) asn1Modules(2)}の下に更に弧が割り当てられている。これらは Rec.X.701 | ISO/IEC 10040 で仕様化されているように、特定の ASN.1 モジュール識別子の登録に対して、次に示すように割り当てられている。

弧	目的
--	----
negotiationDefinitions(0)	抽象構文のネゴシエーションに関する定義を含む、ASN.1 モジュールのバージョン識別子の割り当て

{joint-iso-ccitt ms(9) smo(0) asn1Modules(2) negotiationDefinitions(0)}の下に更に弧が割当てられている。これらは Rec.X.701 | ISO/IEC 10040 で仕様化されたように、特定のバージョンの ASN.1 モジュールの登録に対して、次に示すように割当てられている。

弧	目的
--	----
version1(1)	抽象構文のネゴシエーションに関する定義を含む、ASN.1 モジュールのバージョン 1 を識別する

7.4.2 CMIP に対するオブジェクト識別子の割当て

(Object identifier allocation for CMIP)

<注>CMIP はこれらの弧の割当てに対して責任がある。それらの弧は情報のためだけに含まれている。

CMIP バージョン 1 はいまや旧バージョンであり、バージョン 2 によって取って代わられている。

バージョン 1 は ISO/IEC 9596 で文書化された。これに対応する ITU-T 勧告はない。

{joint-iso-ccitt ms(9) cmip(1)}の下に、各バージョンの CMIP のために、次のように弧が割当てられている。

CMIP 第 1 版

=====

弧	目的
--	----
version1(1)	CMIP 第 1 版に対するオブジェクト識別子の割当て

{joint-iso-ccitt ms(9) cmip(1) version1(1)}の下に、ISO/IEC 9596 に記述されている目的のために、次のように弧が割当てられる。

弧	
--	
aAssociateUserInfo(1)	
aAbortUserInfo(2)	
protocol(3)	
abstractSyntax(4)	

CMIP 第 2 版

=====

弧	目的
--	----
modules(0)	CMIP ASN.1 モジュールに対するオブジェクト識別子の割当て
cmip-pci(1)	CMIP プロトコル制御情報に対するオブジェクト識別子の割当て

{joint-iso-ccitt ms(9) cmip(1) modules(0)}の下に、弧が Rec.X.711 | ISO/IEC 9596-1 に記述されている目的のために、次のように割当てられる。

弧
--
aAssociateUserInfo(1)
aAbortUserInfo(2)
protocol(3)

{joint-iso-ccitt ms(9) cmip(1) cmip-pci(1) }の下に、Rec.X.711 | ISO/IEC 9596-1 に記述された目的のために弧を次のように割当てる。

弧
--
reserved1(1)
reserved2(2)
reserved3(3)
abstractSyntax(4)

7.4.3 機能標準用オブジェクト識別子割当て

(Object identifier allocation for function standards)

{ joint-iso-ccitt ms(9) function(2) }の下に、各機能の勧告／標準を識別するために次のように弧を割当てる。

弧	ドキュメント
--	-----
partX(X)	Rec.X.7NN ISO/IEC 10164-X

ここで、X は ISO/IEC 番号体系に於けるシステム管理機能に対するパート番号であり、X.7NN は対応する ITU-T 勧告番号である。

{joint-iso-ccitt ms(9) function(2) partX(X) }の下に次の弧を割当てる。

弧	目的
--	----
standardSpecificExtension(0)	割当て体系に対する標準固有の拡張
functionalUnitPackage(1)	機能ユニットパッケージ識別子の割当て
asn1Module(2)	ASN.1 モジュール識別子の割当て
managedObjectClass(3)	管理オブジェクトクラス識別子の割当て
package(4)	パッケージ識別子の割当て
parameter(5)	パラメータ識別子の割当て
nameBinding(6)	ネームバインディング識別子の割当て
attribute(7)	属性識別子の割当て
attributeGroup(8)	属性グループ識別子の割当て
action(9)	アクション型の割当て
notification(10)	通知型の割当て

それぞれの機能勧告／標準の範囲内で、このレベル以下に（例えば、特定の属性識別子を割当てるために）、機能勧告／標準の必要性に応じて更に弧が割当てられる。

7.4.4 SMI 標準のオブジェクト識別子割当て

(Object identifier allocation for SMI standards)

{joint-iso-ccitt ms(9) smi(3)}の下に、各 SMI の勧告／標準を識別するために次のように弧を割当てて。

```
弧      ドキュメント
--      -----
partX(X)  Rec.X.72N | ISO/IEC 10165-X
```

ここで X は、ISO/IEC 番号体系に於ける標準のパート番号であり、X.72N は対応する ITU-T 勧告番号である。

{joint-iso-ccitt ms(9) smi(3) partX(X)}の下に次の弧を割当てて。

弧	目的
--	----
standardSpecificExtension(0)	割当て体系に対する標準固有の拡張
asn1Module(2)	ASN.1 モジュール識別子の割当て
managedObjectClass(3)	管理オブジェクトクラス識別子の割当て
package(4)	パッケージ識別子の割当て
parameter(5)	パラメータ識別子の割当て
nameBinding(6)	ネームバインディング識別子の割当て
attribute(7)	属性識別子の割当て
attributeGroup(8)	属性グループ識別子の割当て
action(9)	アクション型の割当て
notification(10)	通知型の割当て

それぞれの機能勧告／標準の範囲内で、このレベル以下に（例えば、特定の属性識別子を割当てるために）、機能勧告／標準の必要性に応じて更に弧が割当てられる。

7.4.5 実クラスに対するオブジェクト識別子の割当て

(Object identifier allocation for actual class)

オブジェクト識別子の値：

{ joint-iso-ccitt ms(9) smi(3) part4(4) managedObjectClass(3) actualClass(42) }

を、Rec.X.720 | ISO/IEC 10165-1 で定義している「実クラス」のセマンティックスを表現するために、この仕様書によって割当てて。CMIS 操作サービス要求に於て基底管理オブジェクトクラスを指定するために用いるときは、このオブジェクト識別子の値はシステム管理操作の受け手がその実クラスのメンバの 1 つとして応答することを表している。

7.5 適合性 (Conformance)

Rec.X.701 | ISO/IEC 10040 に於て、管理情報標準に関する要求条件についての一般的な適合性を明らかにしている。

7.6 管理オブジェクト定義の複雑さ (Complexity of managed object definition)

モデル化の過程を通して、管理オブジェクトの定義に於ける複雑さは最小にすべきである。如何なる場合も、管理操作は関連する OSI エンティティ自身の持つ特性より複雑であってはならない。

7.7 管理オブジェクトの生成と消去 (Managed object creation and deletion)

管理オブジェクトインスタンスの生成と消去は、次のような方法で起こる。

- 管理オブジェクトは管理プロトコルのやり取りによって生成/消去される。対応するセマンティックスとともに、生成および消去の操作はこの目的のために定義される。
- 管理オブジェクトは、それらが関連するリソース、即ち典型的にはプロトコルマシン、への操作の結果として生成/消去される。この場合、生成及び消去の操作は定義されない。1つの例として、コネクションを管理するための表現が挙げられる。
- 管理オブジェクトは他の方法で、生成及び消去されてもよい。この場合、生成及び消去の操作は、その管理オブジェクトに対して定義しない。1つの例は、装置が初期化された時、常に自動的に生成される管理オブジェクトであり、その後、それは管理を通して消去されることがない。

これら管理オブジェクトの生成に関する3つの方法の選択は、管理オブジェクト消去の選択と異なってもよい。

ある場合には、ある特定のクラスの管理オブジェクトの生成及び消去の方法はただ1つの方法しかないことがあり、また別の場合には、これらの方法のうちの1つ以上がありえる。

7.7.1 初期値管理オブジェクト (Initial Value Managed Objects)

管理オブジェクトが生成される時に、のちに管理操作によって変更される対象であっても、生成時に使われるデフォルト値を割当て能力をもたせることが望ましい。

これ（初期値設定）は、初期値管理オブジェクト（IVMO）の仕様によって実現できる。そのIVMOの属性は管理操作によって変更可能であり、IVMOは別のオブジェクトクラスのインスタンス生成で、対応する属性にデフォルト値を与えることができる。

新たな管理オブジェクトがIVMOを使用して生成される場合、IVMO内の属性値がその新しい管理オブジェクト内の対応する属性の初期値として使用される。管理オブジェクトクラスの定義でIVMOがどのように選択されるかを指定する。IVMOの仕様はIVMOが初期値を与える環境、すなわち、どのようにIVMOがその初期値を与えるか及びIVMOが与えるその初期値をどの属性に適用できるかを定義すべきである。

管理操作によってIVMOの属性（値）を変更するとき、それ以前にそのIVMOを使用して生成された管理オブジェクトの属性（値）は、影響を受けない。同じように、IVMOを使用して生成された管理オブジェクトの属性に対する管理操作は、そのIVMOの属性に影響を与えない。

7.7.2 初期属性値のソース (Sources of initial attribute values)

Rec.X.720 | ISO/IEC 10165-1 の定義にあるように、生成時に使用する、管理オブジェクトの初期属性値はいくつかのソースから取得される。ある属性が対応するリソースと一貫性を持たなければならないある特定の値を表現する場合、この値は生成時に使用される必須の初期値となる。

8. 管理オブジェクト定義の一般的原則

(General principles for managed object definition)

ここで示す一般的原則は、管理オブジェクト定義者へのガイダンスを意図したものであり、管理オブジェクト定義間の一貫性を促進させるためのものである。このため、管理オブジェクトの定義者は、可能な限り本ガイダンスを適用することが望ましい。

8.1 共通性 (Commonality)

管理オブジェクトの定義者は、定義の基礎として以下のものを認識し利用することに努めるべきである。

- ITU-T 勧告／国際標準で定義される、共通的な管理オブジェクトクラス；
- Rec. X.721 | ISO/IEC 10165-2 で定義される、共通的な管理オブジェクトクラスと他の特性。

管理オブジェクトの定義者は、定義の共通性を高めるため、他グループの定義を考慮し再利用することに努めるべきである。この目的は、多くのグループの管理オブジェクト定義者に共通な、管理される対象のモデルを開発することによって達成される。

8.2 管理すべきもの (What to manage)

管理オブジェクトクラスとその構成要素の定義は、個々の管理目的に関連する要求条件を満たさなければならない。たとえば、このような要求条件には、レイヤ・サブレイヤの操作における同位プロトコル側面の管理と、サービス境界をまたがった下位のサービス提供者からは特に報告されないが認識可能である諸問題（例えば、許容可能なレベルに達しない、下位サービスにより提供されるサービス品質）がある。管理仕様の開発過程において、個々の管理目的に対し整合性を維持していくことが重要である。管理仕様の各要素（例えば、管理オブジェクトクラス、属性、操作、通知、等）がどのようにこの整合性に関与するか説明するために、コメントを用いるべきである。

管理に関連することは、その事象が発生したリソースを表現する管理オブジェクトを介して記録されなければならない。すなわち、ある特定のリソースを表わす管理オブジェクト（例えば、コネクション）が定義されている場合、そのリソースに関連する情報は、他のものではなく、対応する管理オブジェクトを通じて反映されなければならない。

8.3 構造化 (Structuring)

管理オブジェクトの中での構造を表現するために有効な手法がいくつかあり、これによってデータ、機能のグループを表現することができる。各々の手法には長所・短所があり、ある特定の仕様の要件に対する最適な手法は、以下のいくつかの基準に従って選択される。

Rec. X.720 | ISO/IEC 10165-1 での構造化の手法には以下の物が含まれる。

Rec.X720 | ISO/IEC 10165-1 での構造化の手法には、次のものが含まれる。

- 属性グループ
- サブクラス (特殊化)
- 多重継承
- 包含管理オブジェクト
- パッケージ

属性、操作、及び通知のグループ化は、基本標準における特定のオプションの選択の様に、標準化された条件に基づいて存在するか否かを定義できる。この様な機能に対するグループ化は、全体として存在するか存在しないかのいずれかである。機能のグループ化は、リソースに関するレイヤ勧告／標準に於ける

オプション選択（例えば、トランスポート クラス 4 の規定）の結果、または定義された管理機能（例えば、会計管理）のサポートの結果として生じ、付加的な管理の要求又は能力に結びつく。これら機能のグループ化は、管理オブジェクトクラステンプレートの条件付きパッケージ手法を用いて定義される。

構造化手法の選択におけるひとつの重要な基準は、グループ化が静的であるか動的であるかということである。仮に、そのグループ化が仕様規定時に決定されるものならば、属性グループ、サブクラス、多重継承、または包含管理オブジェクトを使用することが適切である。仮に、そのグループ化の存在が実装時、導入時、あるいはインスタンス生成時に決定されるものならば、包含管理オブジェクトあるいは条件付きパッケージを使用することが適切である。そのグループ化の存在が、包含している管理オブジェクト／カプセル化している管理オブジェクトの存在中に変化するならば、動的に生成・消滅する包含管理オブジェクトを使用することが適切である。

もう一つの規準は、管理オブジェクトの中にグループ化されたインスタンスが複数存在するか否かということである。複数のインスタンスが存在する場合、包含管理オブジェクトの使用が適切である。そうでない場合は 5 つの構造化手法のいずれでも適用できる。

8.4 管理オブジェクト (Managed objects)

8.4.1 スーパークラスのインスタンス生成 (Instantiation of superclasses)

サブクラスを特殊化するもとなる共通基盤を与えるために、インスタンスが生成されることがない管理オブジェクトクラスを定義できる。例えば、汎用 VC (generic virtual circuit) の管理オブジェクトクラスを定義すれば、PVC (permanent virtual circuit) と SVC (switched virtual circuit) がそのサブクラスとなりえる。

ある場合（特にサブクラスが標準を改版するために定義された場合）には、インスタンスが生成されるスーパークラスが存在することがある。

8.4.2 制限を与えないスーパークラス (Unrestricted superclasses)

継承に関する規則により、そのクラスからサブクラスを定義する際、管理オブジェクトクラスの属性の要求値集合と許容値集合を変更する方法が制限される。同様に、この規則はアクション及び通知に対しパラメータを追加する機能を制限する。この制限により、サブクラスがスーパークラスに適合することが保証される。

以上の理由により、将来定義される管理オブジェクトクラスのスーパークラスとなりうる管理オブジェクトクラスを定義する場合、拡張を考慮しておくことが重要である。全ての拡張について予測したり備えたりできなくとも、以下の手法を使用することにより適合性を維持しながら多様な拡張への対応が可能となる。

- 属性値の内のいくつかは早急に必要がないとしても、属性の意味に合致する全ての属性値を含めて、各々の属性の構文（型）を定義する。
- 各々のアクション及び通知の定義に拡張機能を与える。
- より制限されたサブクラスを定義するための基として、これらの要素を含んだ、なんら制限のないスーパークラスを定義する。属性の場合、空の要求値集合と属性構文と等しい許容値集合を意味する。
- 制限の無いスーパークラスの特定のサブクラスを定義し、その属性、アクション、通知に対して必要な制限を加える。

管理オブジェクトの定義者は、制限の無いスーパークラスに関するいくつかの属性、アクション、通知に対してのみ、拡張性を与えてもよい。

8.5 属性 (Attributes)

8.5.1 属性値集合 (Attribute value sets)

ある場合、基本勧告／標準のオプションは、実装時の選択に応じて属性値集合を変更することを許容している。この典型的な例として、基本勧告／標準においては広範なパケットサイズを許容しているが、標準に適合した実装はより制限された範囲のみのサポートで構わない。この場合、属性の振舞い定義で何が可能か示さなければならない。

属性値集合では許容値としてヌル値を定義する必要がある。また、I VMOの属性の場合においては、“対応する属性の値としてヌルを持った管理オブジェクトを生成する”、あるいは“ある初期値のソースとしてこの属性を無視する”というような、特別な意味を割り当てた属性値を定義する必要がある。そのような値を定義する手法には、選択型として抽象構文を定義することを含む。すなわち、ひとつの選択で属性の正常な値集合を定義し、ほかの選択で特別な意味を持った値を定義する。

属性の許容値集合の定義は様々な方法で行われる。例えば、

- 管理オブジェクトクラスの定義の一部として属性値集合を静的に定義すること。
- 属性が含む値集合を示す値を持つ、第2の属性を定義すること。

前者の手法は管理オブジェクトクラスに関する属性定義の数を最小にするが、属性の多様性が要求される場合には、後者の手法の方が値集合の多様さを扱う複数のサブクラスを定義する必要性を回避できる。

8.5.2 属性型 (Attribute types)

構造化属性は、属性構文定義における基本型として順序型、単一型順序型あるいは集合型を用い定義される。これらのASN.1型は単一値属性型に対応するため、属性の要素を個別に変更することが要求されない場合にのみ使用されるべきである。各々を個別に扱う能力を維持しながら、いくつかの属性をまとめて指定する必要がある場合には、属性グループを定義する。必要ならば、グループのメンバ間の関係を明らかにするために、アクションと振舞い定義を使用できる。

注：このことは、属性が個別に扱われた場合には適用されない、属性グループ固有の振舞い定義が存在することを意味する訳ではない。

8.6 属性値の関係 (Attribute value relationships)

ある属性の値が他の属性値のある機能によって制約を受ける場合がある。この種の全ての関係が明らかにされていないなければならない。

ある属性値が同じ管理オブジェクト内の他の属性によって制約を受ける場合、ひとつの管理操作に対して同期に関する要求が存在する。すなわち、関連した属性値の内のいくつかの変更に失敗した場合、結果として関連した属性が不正な値を持つこととなる。このような要求がある場合、管理オブジェクトクラスの振舞い定義の一部として記述されるべきである。

ある属性値が異なる管理オブジェクトの他の属性によって制約を受ける場合で、かつ、同期に関する要求が存在するならば、管理オブジェクトクラスの振舞いに記述されるべきである。この場合で、かつ、管理オブジェクトが同じ被管理システムに全て存在し、単一の管理操作により属性を変更できる場合、同期要求はCMI Sでの分断不可なオブジェクト間の同期機能 (atomic cross-object synchronization capability) を使用することによって実行される。

複数の管理操作間、異なる管理オブジェクトの異なる属性間、あるいは複数の被管理システム間での同期に関する一般的な問題は、現在のシステム管理プロトコルのみで解決できる問題ではない。

8.7 S A Pのモデル化 (Modeling of SAPs)

レイヤに関連した管理オブジェクトの構造の一部として、(N)-レイヤエンティティ、(N)-セクタと(N+1)-レイヤエンティティとの間の関係を表現するための一般的な要求がある。それに対していくつかの解が存在する。例えば、

- (N+1)-レイヤの管理オブジェクトに含まれる情報として関係をモデル化する。
- (N)-レイヤの管理オブジェクトに含まれる情報として関係をモデル化する。
- どのレイヤにも属さない管理オブジェクト、すなわち全てのレイヤに共通な管理オブジェクトに含まれる情報として関係をモデル化する。

本仕様にて推奨する方法は2番目のアプローチである。すなわち、

- (N)-S A Pは各々の管理オブジェクトにより表現されるべきであり、各々の管理オブジェクトは、属性としてアドレス (及びその他の) 情報を持っている。それに加えて、(N)-S A Pに関連する(N)-及び(N+1)-エンティティ管理オブジェクトを指し示す関係属性も持っている。セクタに対する一貫性の要件 (O S Iのアドレス体系を曖昧でないものとするために必要) を実現するために、(N)-S A P管理オブジェクトは、それらがバインドされている(N)-エンティティに対応する管理オブジェクトに包含されなければならない。

注：ここで言われている一貫性とは、(N)-セクタと(N)-エンティティとの組合せで示されるアドレスが、(N+1)-エンティティ（あるいは、同じ型の(N+1)-エンティティ群）をユニークに識別できなければならないということである。この要件は、(N)-エンティティのコンテキストの中で使用される(N)-セクタの割付に対する一意性の要件と同じであるため、このセクタ情報を(N+1)-エンティティではなく(N)-エンティティにて保持すれば、この一貫性の維持はより簡単なものとなる。

8.8 統計 (Statistics)

8.8.1 一貫性 (Consistency)

管理オブジェクトの定義者は、Rec. X.200/ISO 7498 に基づいたいくつかの原則、特に、管理のために必要な情報は、その情報が関係する資源を表現する管理オブジェクトによって記録するという概念を採用することで、レイヤにまたがった統計の一貫性を実現すべきである。

(N)-レイヤにおいて記録される統計項目の候補を以下に示す。

- ローカルエラー
- 同位間での成功した交換
- 同位間での障害
- サービス拒否

例えばこのドキュメントで定義した原則を Rec. X.200/ISO 7498 におけるコネクションの定義に適用すると、主に次のような統計を導ける。

- 他の(N)-レイヤ同位エンティティと確立された(N)-エンティティコネクション数
- (N)-エンティティコネクション確立時におけるローカルな失敗数
- (N)-エンティティコネクション確立時における同位間ネゴシエーションの失敗数
- 下位サービス提供者の(N-1)-コネクション確立拒否数

以上の統計の集合は、（コネクションオリエンテッドの場合）各々のレイヤにおいて発生する事象に関して、カウンタの重複の無い一貫性のある情報与える。

注：同様なモデルがエラー、切断等にも要求される。

8.8.2 PDUカウンタ (PDU counters)

管理オブジェクトの定義者は、(N)-レイヤのSDU（及びSDUオクテット）ではなく、PDU（及びPDUオクテット）のカウンタを規定すべきである。

注：ある限られたいくつかの(N)-レイヤにおいては、PDUオクテット数のみを計数することが必要と考えられている。

8.8.3 重複 (Overlaps)

管理オブジェクトの定義者は、一貫性を実現し、統計上の不要な重複を避けるよう務めなければならない。例えば、最初のPDU送信要求数とリトライPDU送信要求数とを保持している場合、同時に両方のカウンタを増加させる必要はない。これらのカウンタの総和は全送信PDU数となる。

8.8.4 非リセットブルカウンタ (Non-resettable counters)

リセット調整に関連した複雑なインターロッキングメカニズムを必要とせずに複数の観測者を許容するものとして、非リセットブルカウンタが推奨されている。

8.8.5 事象カウンタ (Event counters)

CMISのM-EVENT-REPORTの生成が事象送出ディスクリミネータにより抑止されていたとしても、通知の発行をもたらす管理リソース事象として計数値が更新されなければならない。

8.9 カウンタ (Counters)

カウンタを管理するためには、そのモジュールを知っている必要がある。さもなければ、マネージャはカウンタが一巡する際の値を決定できない。カウンタを定義する場合には、少なくとも次の4つ可能性がある。

- カウンタが決して一巡しない
- モジュールが管理オブジェクトクラス定義の一部として固定されている
- モジュールが関連する属性内で定義される
- モジュールが実装毎に定義され、MOC Sで指定される

注：ISO/IEC JTC1 SC6で定義された管理オブジェクトクラスでは、決して一巡しないカウンタを使用するアプローチが採択されている。

8.10 タイマ (Timers)

各システムは、管理通信において用いられるタイマ属性値を蓄積しなければならないが、その場合、精度に関する共通仕様を守るべきである。これらの属性値とプロトコル上のタイマに対する実際の操作との関係は、振舞いの記述に示される。

注：ISO/IEC JTC1 SC6で定義される管理オブジェクトクラスでは、必要以上の精度ではなく十分広い範囲を包含するために、タイマ値の表現は、16ビット長の仮数部(mantissa)と、最大16ビット長の指数部(exponent)からなる浮動小数点表記が使用される。(このことは、浮動小数点演算の実行の必要性を意味するものではない。)システムは、この精度で値を蓄積出来ると見なされる。他の制限にふれない場合、タイマ属性をこの精度に設定するという要求が受け付けられなければならない。

8.11 属性の更新 (Updating of attributes)

管理オブジェクトクラスの定義者は、管理操作とリソースに対する一般の操作との双方により更新され得る属性を定義する場合、同時に更新される時の影響を明確に定義しなければならない。特に、リソース自身が属性値を更新している場合、同じ属性に対する値の置換操作は無効となることがありうる。

8.12 属性の精度 (Precision of attributes)

管理システムは、エージェントシステムがサポートする精度より高精度の属性値の設定を行っても構わない。そのような高精度の設定値は、規定された精度の近隣値にまるめられる。

8.13 管理オブジェクトの識別 (Managed object identification)

インスタンスが存在する各管理オブジェクトクラスの定義では、その管理オブジェクトに対する名前属性として使用するに適した属性を、少なくとも一つ含まなければならない。この属性は、等しいかどうか試験できる必須属性であり、この属性を命名に使用するという点で、その値は各々の管理オブジェクトの生存期間中は固定した値をとる。この識別子及び値は、同じ包含上位の管理オブジェクトに包含される管理オブジェクトをユニークに識別する。

管理オブジェクトが削除されたならば、その名前属性に割り当てられた値は、同じ包含上位オブジェクトの中で生成された次の管理オブジェクトを識別するものとして、再使用が可能となる。

ある管理オブジェクトクラスの一つのインスタンスが消滅した後も同じクラスの他の全てのインスタンスと識別可能とすることを保証する必要があるならば、管理オブジェクトクラス定義内に、生存期間を超えて一意性が保証される属性（ユニークな識別属性）を定義する必要がある(*1)。この要求が必要ではない管理オブジェクトクラスは、その定義内にユニークな識別属性を含んでいる必要はない。

(*1) 訳注：例えば、タイムスタンプ等が考えられる。

このユニークな識別属性はリードオンリであり、その属性が管理オブジェクト中に含まれる場合は、その管理オブジェクトから発せられる通知の中に含まれるべきである。

8.14 通知 (Notifications)

8.14.1 サービスの拒否 (Denial of service)

下位レイヤのサービスを現わす管理オブジェクトには、サービスの停止となる状況の異常要因を報告する責任があるため、下位のサービスの停止に関する通知は発行されるべきではない。このことは、異常終了が上位レイヤに影響を与えたり、実体の無い通知の生成を禁止することを意図している。

8.14.2 情報の保持 (Information conservation)

通知には、統計情報で保持される情報のみでは失われてしまうかもしれないような、事象に関連した情報が含まれる。例えば：

- プロトコルエラーを検出した受信PDUのヘッダフィールド；
- ある特定の理由で終了しようとしているコネクションに対する統計；
- 一連の特定事象のタイミング；

8.15 操作の使用法 (Use of operations)

管理オブジェクトクラス定義には、適切な操作が含まれなければならない。被管理システムからの要求においては、通知が規定される。管理システムからの要求においては、操作は、被管理システムでの管理オブジェクトへの直接的影響の違いに対応し、以下のように規定される。

- a) 直接的影響が管理オブジェクトクラスのインスタンスの生成となる場合は、生成 (Create) 操作が使用される。他の管理オブジェクトの変化の副次的効果として管理オブジェクトが生成される場合、または、他の管理オブジェクトでの状態変化によりサポート管理オブジェクトが生成される場合のように、いくつかの管理オブジェクト間で整合をとった生成を必要とする複雑な動作には、生成操作は使用されない。
- b) 直接的影響が管理オブジェクトクラスのインスタンスの消去となる場合は、消去 (Delete) 操作が使用される。
- c) 直接的影響が管理オブジェクトの属性値に特定の値を設定することとなる場合は、属性値置換 (Replace attribute value) 操作が使用される。
- d) 直接的影響が管理オブジェクトの属性値に (予め定義されている) デフォルト値を設定することとなる場合は、デフォルト値設定 (Replace with default value) 操作が使用される。
- e) 直接的影響が管理オブジェクトの集合値属性のメンバの追加・削除となる場合は、メンバ追加 (Add member) またはメンバ削除 (Remove member) 操作が使用される。
- f) 直接的影響が管理オブジェクトから属性値を読み出すこととなる場合は、属性値取得 (Get) 操作が使用される。
- g) 他の場合、例えば、直接的影響が全くない場合、直接的影響が上記の組合せである場合、または、全体としてそのオブジェクトになんらかの他の影響がある場合、アクション (Action) 操作が使用される。アクション操作が用いられる場合の例として：
 - 1) 属性値取得、属性値置換、デフォルト値設定、生成、削除、メンバ追加、及び、メンバ削除操作の際に範囲指定とフィルタ機能の手法を用いても、管理オブジェクトの集合に対する要求操作が特定できない場合。
 - 2) 1つ以上の管理オブジェクトをアトミック操作として生成する必要がある場合。
 - 3) 共通な属性を持たないいくつかのオブジェクトに影響が及ぶ場合。
 - 4) 操作における要求または応答が、管理オブジェクトの属性としてモデル化されていない情報を含んでいる場合。

「直接的影響」及び「間接的影響」の概念は、Rec. X.720 | ISO/IEC 10165-1 で論じられている。

9. 管理オブジェクト定義のための記述ツール

(Notation tools for managed object definition)

9.1 記述ツールの概要 (Overview of notational tools)

この節で定義されるテンプレートは、管理オブジェクトクラス定義とそれに関連する命名構造の様々な側面を表現するための、共通ツール群及び共通表記法を提供する。各々のテンプレートの形式定義は 9.3-9.11 に、その形式定義で使用する構文規定は 9.2 に記述する。これらの形式定義により、各テンプレートが持つべきあるいは持つことが可能である構文と、その構文の記述順序とが定義される。これらのツール使用例を、付属資料Aに示す。

管理オブジェクトクラスの構造と振舞いは、まず管理オブジェクトクラステンプレート (Managed Object Class Template) によって定義される。このテンプレートは、管理オブジェクトクラスと他の管理オブジェクトクラス間に存在する継承関係と、管理オブジェクトクラスの中に定義されている振舞い、属性、通知、操作のパッケージとを明らかにする。他の管理オブジェクトクラスの定義を行う際にこの定義の部分的再使用を可能とするため、属性 (個別、グループ)、振舞い、アクション、通知、パラメータ、パッケージの各定義を提供する付加テンプレートが規定されている。これらのテンプレートは、他のテンプレートから 9.2 で定義する参照メカニズムにより呼び出される。このメカニズムは、ひとつの標準から、他の標準に含まれるような規定への参照を可能とする。それゆえ、ローカル規定につけ加えて、一般規定が管理オブジェクトクラス定義の中で使用可能となる。これらテンプレートは、必要ならばインラインの形で含まれていても構わない。

管理オブジェクトクラスの命名は、ネームバインディングテンプレートで定義される。このテンプレートは命名される管理オブジェクトクラスを識別し、ある特定の包含上位クラスのコンテキストにおいて、そのクラスのインスタンスを命名するために使われる相対識別名を定義する。更にこのテンプレートは、ある特別なネームバインディングの結果として、二つの管理オブジェクトクラス間に存在する関係の規定も提供する。

9.2 テンプレート定義で用いられる規約

(Conventions used in template definitions)

テンプレートは、`template-label` と `TEMPLATE-NAME` から始まる。テンプレートは、ひとつあるいは複数の構造体を含み、その各々は `CONSTRUCT-NAME` によって名付けられ、各々がひとつの `construct-argument` を持つことができる。`construct-argument` は、個々の構造体定義の要求に従い、複数の要素を持つことができる。あるテンプレートを使用する各々のインスタンスは、他のテンプレートからそのインスタンスが参照可能となるようなユニークな `template-label` を宣言する。そして、`REGISTERED AS` 構造体が存在する場合、そのインスタンスが登録された `ASN.1` オブジェクト識別子の値を割り当てる。各々の構造体の終わり (`REGISTERD AS` と `DEFINED AS` を除いて) の印、及びテンプレートの終わりの印として、文字セミコロンが用いられる。

テンプレートの構造を単純化するために、例えば同じ構文の構造体が同一テンプレート定義の中で繰り返される場合等、補助構文定義を用いてもよい。テンプレート定義を完成させるため補助構文定義が要求される場合は、テンプレート定義の最後に次のキーワードが用いられる。

`supporting productions`

続いて、次の形式を持ったいくつかのプロダクションが記述される。

`<definition-label> -> <syntactic-definition>`

definition-label は、その定義をテンプレート定義あるいは他の supporting productions から参照されることを可能としており、また、syntactic-definition では、本副節の残りのところで定義されている構文上の規約を用いて定義の拡張を行う。いくつかの選択的の文字列を定義する syntactic-definition の場合、それを含んでいる supporting production への参照は、選択リストの中から選ばれた一つの文字列に対して評価される。

テンプレート定義は、次のような構文上の規約に基づいている。

- a) テンプレート定義の一部分を成す全ての終端記号及びキーワードにおいては、大文字小文字が区別される。
- b) テンプレートの構文を曖昧でないものとするために、必要に応じ、テンプレートの各々の要素は、一つ以上のデリミッタで隣接の要素と分離されなければならない。有効なデリミッタは、スペース、行の終わり、ブランク行、及びコメントである。一つ以上のデリミッタを以下の相互間におく必要がある。

- 1) template-label と TEMPLATE-NAME の間
- 2) TEMPLATE-NAME と CONSTRUCT-NAME の間
- 3) CONSTRUCT-NAME と construct-argument の間

これ以外に、一つのテンプレート中のどの要素ペア間に、一つ以上のデリミッタをおいても構わない。そして、construct-argument が多くの別々の要素から構成される場合、それらの間にデリミッタがおかれても構わない。その他の場合は、テンプレート定義がデリミッタの挿入を明示的に許容していない限り、デリミッタはテンプレートの要素間に挿入されるべきではない。

- c) スペース、ブランク行、コメント、及び行の終わりは、デリミッタとしてのみ意味を持つ。
- d) コメントは、次の文字ペアで始まり、

--

次の文字ペアもしくは行の終わりの何れか先に現れた方で終了する。

--

本仕様で規定されるテンプレートの解釈を行う上で、コメントはスペースと同等である。コメントは特に標準的な意味を持たない。

- e) 文字

;

は、テンプレート中の各々の構造体の終了（REGISTERD AS と DEFINED AS を除いて）、及びテンプレートそのものの終了を示すために用いられる。

- f) オブジェクト識別子を表現する記法は、Rec. X.208 | ISO/IEC 8824 で定義されるオブジェクト識別子値を表現する値表記法である。例えば、プロダクション、

object-identifier -> <ObjectIdentifierValue>

は、このドキュメント中の全テンプレート定義に対する補助プロダクションである。ここで ObjectIdentifierValue は、Rec. X.208 | ISO/IEC 8824 で定義されている該当記法を参照する。

g) 次の、

[]

で囲まれた文字列は、テンプレート定義の中で、テンプレートを使用する各インスタンスには存在しても存在しなくてもよい部分の範囲を示す。

次のように、鍵括弧にアスタリスクがついている場合、

[]*

括弧内の内容は現れなくとも良いし、何度現われても良い。定義の中のこれらの部分が省略されたり、繰り返されたりすることは、テンプレート型の定義に依存する。

h) 次の、

<>

で囲まれた文字列は、テンプレートを使用する各インスタンスにおいて、置換される文字列の範囲を示す。置換文字列の構造と意味は文字列型に依存する。

i) 大文字の列は、テンプレートを使用する各インスタンス中に存在しなければならないキーワードを表わす。ただし、それらは以下の、

[]

に囲まれているオプションとなる場合を除く。

j) 文字

|

は、supporting production の syntactic-definition の中の選択文字列を区切るデリミッタとして使用される。supporting production が選択文字列を定義するために使われる場合、最初の選択子の開始デリミッタは -> であり、| は選択子の終了デリミッタであると共に、次に続く選択子の開始デリミッタである。最後の選択子の終了デリミッタは、最終選択子の開始デリミッタの次に現れる行の終わりである。

k) **template-label** は、それが定義されている勧告／標準またはドキュメントの中でユニークでなければならない。マルチパートとなる勧告／標準またはドキュメントでは、通常は個々のパートが独立して維持、出版される。その場合、**template-label** は、それが定義されているパートの中でユニークでなければならない。

template-label の一意性に対する要求は、ラベル付けされるテンプレートの型とは独立である。例えばあるドキュメント内で、**label1** という文字列が、あるテンプレートを用いたインスタンスへのラベルとして用いられている場合、同じドキュメント内で、同じ型または他の異なる型のテンプレートを用いた他のインスタンスへのラベル付けのために、同じ **label1** という文字列を用いてはならない。

ある **template-label** がドキュメントAで定義され、ドキュメントBで参照される場合、ドキュメントBにおける参照においては、ドキュメントAのグローバルでユニークな名前をプレフィックスとして用いる必要がある。ITU-T または ISO/IEC のような国際的に認知された命名機関によって名前付けされたドキュメントの場合には、"Rec. x.722 | ISO/IEC 10165-4:1992"のように、登録された英数字からなるドキュメント名が、識別子として使用される。この文字列の形式は、命名機関によって指定された形に従わなければならない。参照ドキュメントが ITU-T と ISO/IEC により共同で開発／出版されている場合、ドキュメント名の識別子は上で示したように、ITU-T と ISO/IEC のドキュメント名識別子を、“|”により区切って列記する形となる。グローバルでユニークな名前がまだ有効でない場合は、参照ドキュメントにオブジェクト識別子の値を割り当て、その値をグローバルでユニークなドキュメント名として使用することが許される。**template-label** の構文は、上記の記法を用いて以下のように定義される。

[document-identifier :] <label-string>

document-identifier -> "<standard-name>" | object-identifier

label-string は、任意数の次のキャラクタから成る。

- 1) 大文字小文字のアルファベット；
- 2) 数字 0-9；
- 3) 文字
-
/
- 4) 文
/

label-string は、小文字のアルファベットで始まり、上記の文字はどのような順序で現れても構わない。例外として、次の文字ペア、

--

は、label-string の中には現れてはならない。例えば、次の template-label、

"Rec. X.722 | ISO/IEC 10165-4:1992" : exampleObjectClass

は、付録Aに含まれる exampleObjectClass の定義に対するグローバルでユニークなラベルとなる。

Document-identifier がプレフィックスとして用いられていないラベル参照は、その参照が現われるドキュメント内で宣言されたラベルへの参照とみなされる。

- l) あるテンプレートに他のテンプレートへのポインタとして template-label が存在する場合は常に、参照されているテンプレート自身（テンプレートのラベルを含む）のテキスト全体で置き換えられても良い。このことは、あるテンプレートがそれが参照しているテンプレート（サブテンプレート）をインラインとして含むことを可能とし、他のテンプレートがそのサブテンプレートを参照する能力も保持される。例えば、次の補助プロダクション、

template-label -> template-definition

は、template-label と template-definition からなる全てのインスタンスを表すものとみなされる。

- m) テンプレートから A S N. 1 の型あるいは値定義への参照が必要な場合は常に、A S N. 1 の型または値の名前の前に、その型または値の定義が含まれている A S N. 1 モジュールのモジュール名が置かれる。そのモジュール名は、型または値の参照を行っているテンプレートと同じドキュメント内にある A S N. 1 モジュールを、参照するものとみなされる。以下の補助プロダクション、

type-reference -> <module-name>.<type-name>

value-reference -> <module-name>.<value-name>

は、従って A S N. 1 の型あるいは値を参照する全てのテンプレート定義を表すものとなる。ここで module-name は、このドキュメント内に含まれ、参照される A S N. 1 モジュールに割り当てられた名前であり、type-name 及び value-name は、そのモジュール内に含まれる A S N. 1 の型または値の定義に割り当てられた名前である。他のドキュメントに含まれる型または値の定義を参照する必要がある場合、ローカル A S N. 1 モジュールにおいて A S N. 1 の IMPORT 機構を用いることにより、適切な型または値の定義の取り込みが可能となる。

n) テンプレートにテキストを含む必要がある場合は常に、テキストは文字列の形式で挿入され、オプションとして次に示す文字の中から選択した `text-delimiter` により開始、終了する。

! " # \$ % ^ & * ' ` ~ ? @ ¥

{/* P. S. ¥: バックスラッシュ \ */}

`text-delimiter` 文字が用いられた場合、開始と終了は同じ文字を用いなければならない、その `text-delimiter` 文字が文中で使用される場合は常に、その文字が2回繰り返される形に置き換えられる。`text-delimiter` 文字が用いられない場合、構文上、テキストが使用されているテンプレート定義内では、テキスト列の中にテンプレートのデリミタとして用いられる文字を含んではならない。

補助プロダクション、

```
delimited-string -> text-delimiter <text-string> text-delimiter |
                    <text-string>
```

`text-delimiter` -> ! | " | # | \$ | % | ^ | & | * | ' | ` | ~ | ? | @ | ¥
{/* P. S. ¥: バックスラッシュ \ */}

は、従って `delimited-string` が使用可能である全てのテンプレートとみなされ、ここで、`text-string` は任意の文字の順序列である。`text-delimiter` が用いられた場合、`text-string` 中に現れるその区切り文字は全て、1対の(2個の) `text-delimiter` 文字に置き換えられる。

区切り文字の使用に関する規則の例外として、`text-string` の内部構造の場合、特に本仕様に定義されるコメント構造の使用の場合には、本仕様の規定に従う必要はない。従って、本記法を適用するドキュメントの中で特に明示的な記述がなければ、`text-string` 全体の意味は標準的な意味を持つものとみなさなければならない。

(注) 管理情報定義を行う上で日本語の記法が必要となるが、ASN.1 の日本語拡張記法として、以下の記述方法が JIS (日本工業規格) X5603”開放型システム間相互接続の抽象構文記法 1 (ASN.1) 使用”で定められており、本仕様は、これに基づくものとする。

- (1) ASN.1 使用の文字集合に JIS X0208 (情報交換用漢字符号体系) で規定する図形文字のうち、漢字 (第一水準、第二水準)、平仮名、片仮名、長音記号、及びアンダラインを追加する。
- (2) 日本語拡張記法を使用した場合の、ASN.1 項目の先頭文字の大文字/小文字の区別は、次の方法による。

(1)で追加したアンダライン以外の文字はすべて小文字として扱い、大文字で始めるべき ASN.1 項目に日本語を使用する場合には、先頭文字を英大文字またはアンダライン文字とする。

次に例を示す。

例：(基本記法) `Abstract-syntax-name ::= OBJECT IDENTIFIER`

(日本語拡張記法) `__抽象構文名 ::= OBJECT IDENTIFIER`

- (3) 次の ASN.1 に対しては、日本語拡張記法を規定しない。

- ・文字列型、有用型の型参照
- ・マクロ参照

9.3 管理オブジェクトクラステンプレート (Managed object class template)

9.3.1 概要 (Overview)

管理オブジェクトクラステンプレートは、管理オブジェクトの形式的定義の基となるものである。テンプレートの要素を用いて、クラスを継承木の適切なノードに配置したり、クラスを持つさまざまな特性や振舞いを定義することができる。定義の主要要素を以下に示す。

9.3.1.1 継承 (Inheritance)

各管理オブジェクトクラスでは、自らが導き出されるスーパークラス (複数) を定義する。スーパークラスの特性はサブクラスに継承される。サブクラスの定義では、これらの特性に追加 (特殊化) を行うことはできるが、スーパークラスの如何なる特性も削除はできない。全てのクラスは `top` のサブクラスである。

9.3.1.2 必須パッケージ (Mandatory packages)

管理オブジェクトクラス定義は、そのクラスの全てのインスタンスを特徴づける性質を完全に指定する振舞い/属性/操作/通知のパッケージを含む。

9.3.1.3 条件付きパッケージ (Conditional packages)

管理オブジェクトクラス定義は、指定された条件が満足されれば、そのクラスのインスタンス内に存在する振舞い/属性/操作/通知のパッケージを含む。

9.3.1.4 クラスの名前付け (Class naming)

管理オブジェクトクラス定義には、管理プロトコルの中でクラスを参照するために用いられるクラス名を含まなければならない。クラスの名前付けは、その管理オブジェクトクラス定義を示すオブジェクト識別子の値を登録することにより完結する。

9.3.2 テンプレート構造 (Template structure)

<class-label> MANAGED OBJECT CLASS

```
[DERIVED FROM      <class-label>    [,<class-label>]*;
]
[CHARACTERIZED BY  <package-label> [,<package-label>]*;
]
[CONDITIONAL PACKAGES <package-label> PRESENT IF condition-definition
                        [<package-label> PRESENT IF condition-definition]*;
]
```

REGISTERED AS object-identifier;

supporting productions

condition-definition -> delimited-string

9.3.3 サポートする定義 (Supporting definition)

DERIVED FROM <class-label> [,<class-label>]*

DERIVED FROM 節は、top 以外の全ての管理オブジェクトクラスの定義内に存在しなければならない。それ故、top は自分以外の全てのクラスのスーパークラスである。

class-label には、その由来元である管理オブジェクトクラス、すなわちその直上のスーパークラス (の一つ) を指定する。多重継承が可能であるため、一つ以上の直上のスーパークラスが存在してもよい。

継承 (特殊化) の過程では、サブクラスの定義内にスーパークラスの全特性が含まれなければならない。

スーパークラスから継承された特性はサブクラスの記述内で反復してはならない。但し、スーパークラスから継承された定義を拡張、変更するための記述法 (本仕様で示している) を用いる場合はこの限りでない。従って、DERIVED FROM 節はスーパークラスの定義から自動的に全特性を取り込むものとして扱われる。これらの特性は、CHARACTERISED BY および CONDITIONAL PACKAGES の各節内で定義された要素によって追加、変更され得る。

注) 管理オブジェクトクラスの定義は、その特性を継承する元になった全ての管理オブジェクトクラスのリストを、コメントとして、その定義内に含まなければならない。

多重継承の結果、同じ要素の定義が重複して取り込まれた場合 (例えば、二つのスーパークラスの定義が同一の属性を含む場合に起こり得る)、サブクラスは関係する定義を一つだけコピーして持つものとする。

特殊化により、オブジェクトクラスの定義内に継承された、またはその中に含まれるパッケージと条件付きパッケージの要素間には不整合が生じ得る。これらを解決するために、ある特定のオブジェクトクラスに含まれる全てのパッケージは同一のレベルのものとして扱われる。各パッケージはいくつかの要素を定義しており、それらは以下のように扱われる。

a) BEHAVIOR.

サブクラス内に含まれるパッケージで継承された振舞いを拡張する。管理オブジェクトクラスの振舞いは、条件付きパッケージの有無を考慮した形で表現されなければならない。

注) ある環境においては、スーパークラスから継承されたもの以外にまったく付加的な振舞い定義を必要とせずに、サブクラスが定義され得る。

b) ATTRIBUTES.

属性はサブクラス定義に含まれるパッケージ内で指定できる。パッケージの ATTRIBUTE 節が、管理オブジェクトクラス内で重複して定義された属性を指定する場合は、次の規則が適用される。

- 1) 実際の管理オブジェクトインスタンスには、その型の一つの属性のみが含まれる。
- 2) 結果として生ずる `propertylist` は、サブクラス内に含まれる `propertylist` と、継承された `propertylist` の論理的和となる。但し、PERMITTED VALUES (許容値) と REQUIRED VALUES (要求値) については例外である。すなわち、許容値の場合、実際のインスタンスの許容値は、その属性型に対する全ての許容値定義 (複数) の共通集合となる。要求値の場合、実際のインスタンスの要求値は、その属性型に対する全ての要求値定義 (複数) の和集合と、実際のインスタンスの許容値集合の共通集合となる。DEFAULT VALUE (デフォルト値) や INITIAL VALUE (初期値) については、複数の定義内でその属性に対して矛盾する値が指定されている場合、サブクラスに含まれるパッケージが、その矛盾を解決しなければならない。
- 3) ある属性に関するパラメータは、属性テンプレート内の全てのパラメータと、インスタンス生成された全てのパッケージ内の属性に関する全てのパラメータの積集合となる。

インスタンス生成される管理オブジェクトクラスの場合は、クラス定義の一部として定義された属性が少なくとも一つ、存在しなければならない。これは管理オブジェクトインスタンスを命名するために用いられる属性を指定する必要があるからである。

注) 命名のための属性はクラス定義内のどの属性からも選択可能である。これにはスーパークラスから継承された全ての属性と、特殊化の過程で付加されたすべての属性を含む。

c) ATTRIBUTE GROUPS.

拡張可能な属性グループの場合、サブクラスのインスタンス内の属性グループの内容は、属性グループテンプレートで定義された全ての属性と、スーパークラスまたはサブクラス内でそのグループに付け加えられた全ての属性を加えたものである。

d) ACTIONS.

アクションはサブクラス定義内に含んでもよい。それらはスーパークラスから継承されたアクションに対しさらにアクションを追加したもの、または継承されたアクションに対しパラメータを追加したものである。あるアクションに関するパラメータはアクションテンプレートに関する全てのパラメータと、インスタンス化されたパッケージ内のアクションに関する全てのパラメータの和集合である。

e) NOTIFICATIONS.

通知はサブクラス定義内に含んでもよい。それらはスーパークラスから継承された通知に対しさらに通知を追加したもの、または継承された通知に対しパラメータを追加したものである。ある通知に関するパラメータは通知テンプレートに関する全てのパラメータと、インスタンス化されたパッケージ内の通知に関する全てのパラメータの和集合である。

あるパッケージが管理オブジェクトクラスの定義内に一度以上含まれる場合 (継承あるいは管理オブジェクトクラステンプレート内での多重参照によりもたらされる)、結果として生ずる、パッケージに関する `condition-definition` は、複数の定義内の全ての `condition-definition` の論理的和となる。この目的のため、CHARACTERISED BY 節に含まれるパッケージ (必須パッケージ) は、あたかも PRESENT IF !TRUE! の条件定義をもつ CONDITIONAL PACKAGE 節に含まれたものとして扱われる。

あるパッケージ（必須あるいは条件付き）の特性は、そのパッケージを含む全ての管理オブジェクト内に必要な特性が存在することを、他の条件付きパッケージの条件が保証するならば、その条件付きパッケージの特性にのみ依存する。

（訳注：必須パッケージの場合もこの条件を満足するかは疑問）

CHARACTERIZED BY <package-label> [,<package-label>]*

この節は、DERIVED FROM 節の結果として定義の一部となるものに加え、一つ以上の必須パッケージ（振舞い、属性、操作、通知）を管理オブジェクトクラス定義の中に含むことを可能とする。package-label には含まれるべきパッケージ定義を指定する。条件付きパッケージとしてその管理オブジェクトクラス定義に含まれるパッケージのラベルを指定することにより、そのパッケージを当該管理オブジェクトクラスおよびそのサブクラスの中で必須とすることができる。

CONDITIONAL PACKAGES <package-label> PRESENT IF condition-definition
[,<package-label> PRESENT IF condition-definition]*

一つ以上の条件付きパッケージがそのクラスに含まれる場合に指定する。package-label には対応するパッケージ定義を指定する。condition-definition には、もしそれが真ならばそのパッケージがそのクラスのインスタンスに含まれることを要求する条件を記述する。condition-definition を表現するのに用いられる構造や文字集合に、なんら制限はない。その条件は、Rec.X.720 | ISO/IEC 10165-1 で示される条件付きパッケージの要件を満たさなければならない。例えば、

CONDITIONAL PACKAGES class-4-attributes PRESENT IF 'the corresponding entity supports Class 4 operation as specified in ISO/IEC XXXX';

は、ISO/IEC XXXX がクラス 4 オペレーションを資源の正当なオプションフィーチャとして定義しているとすれば、正しいパッケージ宣言を構成している。

REGISTERED AS <object-identifier>

object-identifier の値は、オブジェクトクラス定義に対するグローバルにユニークな識別子を提供する。この値は管理プロトコルの中で、オブジェクトクラスを識別する必要が生じた時に用いられる。

9.4 パッケージテンプレート (Package template)

9.4.1 概要 (Overview)

このテンプレートによって振舞い定義、属性、属性グループ、操作、通知およびパラメータの組合せにより構成されるパッケージが定義できる。これらのパッケージは、CHARACTERIZED BY 節または CONDITIONAL PACKAGES 節を用いて、後続する管理オブジェクトクラステンプレート内へ取り込まれる。主要要素を以下に示す。

9.4.1.1 振舞い (Behavior)

パッケージ定義では、そのパッケージに含まれる振舞いの完全な指定を行う。これには、以下のものが含まれる。

- 管理オブジェクトに対する操作の効果、および通知が発行される環境条件。
- 一貫性の規則を満たすために操作に課せられる制約、特に管理オブジェクトを生成、消去する場合の規則とそれらの操作の結果。
- ある管理オブジェクトクラスのインスタンスと関係する、同じまたは異なるクラスの他の管理オブジェクトとの相互作用の指定。
- もし存在する場合、通知内の情報に関する属性の識別。これには、その属性と特定の通知フィールドとのマッピング、および、通知の発行条件との関係の指定を含む。
- もし存在する場合、初期値管理オブジェクト (IVMO) の選択基準の指定。
- 管理オブジェクトクラスの振舞いの、その他の側面の完全な定義。

9.4.1.2 包含される属性 (Contained attribute)

パッケージが含む属性の集合が指定されなければならない。

9.4.1.3 操作と通知 (Operations and Notifications)

パッケージ定義では、次のことを指定しなければならない。すなわち、このパッケージを利用するクラスのインスタンスが生成できる通知、実行できる操作、および属性に関する操作の場合はその操作の対象となる属性。さらに、このパッケージを利用する管理対象クラスのインスタンスの通知および操作に含まれる全ての付加パラメータ。

注1) クラス定義内で示される操作は、Rec.X.720 | ISO/IEC 10165-1 で定義された操作の型 (属性値取得 (Get attribute value)、属性値置換 (Replace attribute value)、デフォルト値設定 (Replace with default value)、等) である。アクションおよび通知の場合は、9.10、9.11 に示すように、それらの機能を特徴付けるために、追加の定義が必要である。生成および消去操作は、9.6 のネームバインディング・テンプレートの一部として指定される。これは、管理オブジェクトの生成、消去が、管理オブジェクトクラスのインスタンスというより、上位および下位のオブジェクト間の包含関係に密接に関連するためである。

注2) レートバインディング (すなわち、管理オブジェクトクラスのアクションや通知に対する付加パラメータの割当) は、関係するアクションや通知およびそれらの新パラメータ (のみ) を含むパッケージを、管理オブジェクトクラス内に含むことにより実現できる。アクションおよび通知のパラメータに関する 9.3.3 節の積集合の規則は、パッケージがインスタンス化される時に、付加パラメータがアクションや通知に関連付けられることを意味する。

9.4.2 テンプレート構造 (Template structure)

<package-label> PACKAGE

```
[BEHAVIOR      <behavior-definition-label> [,<behavior-definition-label>]*;
]
[ATTRIBUTES   <attribute-label> propertylist [<parameter-label>]*
              [,<attribute-label> propertylist [<parameter-label>]*]*;
]
[ATTRIBUTE GROUPS <group-label> [<attribute-label>]*
              [,<group-label> [<attribute-label>]*]*;
]
[ACTION       <action-label> [<parameter-label>]*
              [,<action-label> [<parameter-label>]*]*;
]
[NOTIFICATIONS <notification-label> [<parameter-label>]*
              [,<notification-label> [<parameter-label>]*]*;
]
[REGISTERED AS  object-identifier];
```

supporting productions

Propertylist -> [REPLACE-WITH-DEFAULT]

```
[DEFAULT VALUE   value-specifier]
[INITIAL VALUE   value-specifier]
[PERMITTED VALUES type-reference]
[REQUIRED VALUES type-reference]
[get-replace]
[add-remove]
```

value-specifier -> value-reference

| DERIVATION RULE <behavior-definition-label>

get-replace -> GET | REPLACE | GET-REPLACE

add-remove -> ADD | REMOVE | ADD-REMOVE

9.4.3 サポートする定義 (Supporting definitions)

BEHAVIOR <behavior-definition-label> [,<behavior-definition-label>]*

BEHAVIOR 節では、パッケージに関する振舞い (の意味) を完全に記述する。この節は管理オブジェクトの外部からみた側面 (操作および通知) をその内部操作に関連付ける。behavior-definition-label には、BEHAVIOR テンプレートをを使用した定義のラベルを指定する。ある環境では、パッケージは振舞い定義無しで定義され得る。

ATTRIBUTES <attribute-label> propertylist [<parameter-label>]*
[,<attribute-label> propertylist [<parameter-label>]*]*

この節ではパッケージの属性を記述する。各 attribute-label に続く propertylist には、その属性に対し管理オブジェクトに行われる一連の操作を定義する。更に、その属性に関するデフォルト値、初期値、許容値、要求値を定義する。

属性がデフォルト値置換操作によって設定されるデフォルト値をもつ場合、REPLACE-WITH-DEFAULT を指定する。

属性がデフォルト値をもつ場合、DEFAULT VALUE を指定する。これは、デフォルト値置換操作時の属性値を提供するため、または Rec.X.720 | ISO/IEC 10165-1 で定義された規則に従ってパッケージがインスタンス化された時にその属性に用いられるデフォルト値を指定するためである。デフォルト値が指定されず、かつ REPLACE-WITH-DEFAULT が存在する場合、デフォルト値は被管理システムにおけるローカルな他の手段によって決定される。デフォルト値は、value-reference、またはデフォルト値を決定する方法を示す DERIVATION RULE によって指定される。

属性が必須の初期値をもつ場合、その属性の初期値を提供するために生成時に用いられる INITIAL VALUE を指定する。初期値は、value-reference、または初期値を決定する方法を示す DERIVATION RULE によって指定される。

PERMITTED VALUES が存在する場合、type-reference には属性が取り得る値に関する制約を指定する。その形式は、ASN.1 サブタイプ記法を用いて定義される属性シンタックス型のサブタイプでなければならない。

注) PERMITTED VALUE 節は、属性シンタックスの仕様で許された値の集合に対する制約を指定する必要がある (例えば、現在の属性仕様を変更する時) 属性定義においてのみ指定する。属性値に対するそのような制限は、合理的な値の集合とみなした任意の仮定にもとづく制限ではなく、それが属性のセマンティクスにまで影響する場合にのみ行われなければならない。

REQUIRED VALUES が存在する場合、type-reference には属性が取るべき値を指定する。その形式は、ASN.1 サブタイプ記法を用いて定義される属性シンタックス型のサブタイプでなければならない。

注) この文は、適合性を保つために必要な値の集合を定義する。例えば、モデム管理オブジェクトは、ボーレート属性として 0-19.2K の値を取り得るが、モデム標準に対する適合性としては、許された値の範囲内である特定の値のサポートを必要とする。PERMITTED VALUE に関しては、ある属性に対しサポートされる値集合の要求は次に示す場合にのみ行われるべきである。すなわち、ある属性の値に関する要求が、合理的な最小の値集合とみなした任意の仮定にもとづく要求ではなく、属性のセマンティクスからの特性に基づく場合のみである。

parameter-label には、属性に対する管理操作についての管理オブジェクトクラス固有のエラーパラメータを指定する。これらは、処理障害 (processing failures) として報告される。エラーパラメータのシンタックスは参照するテンプレートで定義される。

ATTRIBUTE GROUPS <group-label> [<attribute-label>]*
[,<group-label> [<attribute-label>]*]*

この節は、属性グループの集合をパッケージの一部として指定する。拡張可能属性グループの場合は、追加の attribute-label を指定することにより元の定義を拡張できる。

ACTIONS <action-label> [<parameter-label>]*
[,<action-label> [<parameter-label>]*]*

action-label には、パッケージ内に含まれるアクション定義を指定する。振舞い定義では、管理オブジェクトに対するこれらのアクションの効果を指定しなければならない。

parameter-label には、管理オブジェクトクラス固有のアクション情報や応答パラメータ、またはこのアクションに対応する管理オブジェクトクラス固有のエラーパラメータを指定する。パラメータのシンタックスは参照するテンプレートで定義される。

NOTIFICATIONS <notification-label> [<parameter-label>]*
[,<notification-label> [<parameter-label>]*]*

任意の通知をパッケージ内に含む場合に指定する。notification-label には、適用する通知の定義を指定する。振舞い定義では、管理オブジェクトがこれらの通知を生成する環境を指定しなければならない。

parameter-label には、管理オブジェクトクラス固有のアクション情報や応答パラメータ、またはこの通知に対応する管理オブジェクトクラス固有のエラーパラメータを指定する。これらは、付加的なパラメータとして (例えば、X.733 | ISO/IEC10164-4 (警報報告機能) で定義されている通知の problemData フィールドとして) 使用される。パラメータのシンタックスは参照するテンプレートで定義される。

REGISTERED AS <object-identifier>

object-identifier 値は、パッケージ定義のグローバルでユニークな識別子を提供し、そのパッケージで定義する振舞い定義、属性、属性グループ、アクションおよび通知の集合を登録する。object-identifier の値は、管理オブジェクトクラスから生成されるインスタンスの Packages 属性に含まれる値である (X.720|ISO/IEC 10165-1 に述べられた規則に従う)。この節は、パッケージが管理オブジェクトクラステンプレート内の CONDITIONAL PACKAGES 節から参照される場合に必要となる。

9.5 パラメータテンプレート (Parameter template)

9.5.1 概要 (Overview)

このテンプレートは、パラメータシンタックスの指定および登録を行う。また、9.4、9.7、9.10、9.11 で定義された PACKAGE, ATTRIBUTE, ACTION, NOTIFICATION の各テンプレート内の個々の属性、操作、および通知に関連する振舞いの指定および登録を行う。パラメータテンプレートで指定された型は、管理 PDU 内の ANY DEFINED BY X 節を埋めるために用いられる。ここで、x はパラメータに割り当てられたオブジェクト識別子を運ぶ PDU 内のフィールドである。このメカニズムは、例えば、以下の定義に適用される。

- － 処理障害
- － アクション要求／応答のパラメータ
- － 通知要求／応答のパラメータ

これらの各構文内のテンプレートの使用は、9.5.3 で述べられる。

定義の主な要素を以下に示す。

9.5.1.1 コンテキスト定義 (Context definition)

このテンプレートは、パラメータが適用されるコンテキストを指定する。すなわち、パラメータが管理 PDU 内の特定フィールドで運ばれることを指定する。

9.5.1.1.1 アクション情報／応答、イベント情報／応答、特定エラー

(Action Info/Reply, Event Info/Reply, Specific Error)

コンテキストが、パラメータを運ぶ管理 PDU によって明確に識別される時、そのコンテキストは、9.5.3 で指定された 5 つのキーワードのいずれかで示される。コンテキストは、ANY DEFINED BY 節が、管理 PDU 内にただ一度だけ現れる場合にのみ、その PDU によって明確に識別される。

9.5.1.1.2 コンテキストキーワード (Context Keyword)

コンテキストが、パラメータを運ぶ管理 PDU によって明確に識別されない場合、コンテキストキーワードが指定されなければならない。コンテキストキーワードは、パラメータを運ぶ管理 PDU のフィールドを識別しなければならない。

9.5.1.1.3 他のテンプレートにおける使用 (Usage in other template)

表 1 に、パラメータテンプレートが参照される場所と許されるオプションを示す。

表 1. パラメータテンプレートの使用

使 用	可能なコンテキスト
PACKAGE テンプレートの ATTRIBUTE 節	SPECIFIC-ERROR
PACKAGE テンプレートの ACTION 節	Context-keyword SPECIFIC-ERROR ACTION-INFO ACTION-REPLY
PACKAGE テンプレートの NOTIFICATION 節	Context-keyword SPECIFIC-ERROR EVENT-INFO EVENT-REPLY
NAME BINDING テンプレートの CREATE 節	SPECIFIC-ERROR
NAME BINDING テンプレートの DELETE 節	SPECIFIC-ERROR
ATTRIBUTE テンプレート	SPECIFIC-ERROR
ACTION テンプレート	Context-keyword SPECIFIC-ERROR ACTION-INFO ACTION-REPLY
NOTIFICATION テンプレート	Context-keyword SPECIFIC-ERROR EVENT-INFO EVENT-REPLY

パッケージ定義内の修飾子として用いられる場合は、パラメータはそれが修飾している要素にレートバウンド (late bound) される。例えば、通知シンタックスが拡張可能な場合、パッケージ定義の時点で、以前に定義された通知にパラメータが追加され得る。

9.5.1.2 シンタックス定義 (Syntax definition)

テンプレートにより、抽象構文がパラメータと関連づけられる。

9.5.1.3 属性参照 (Attribute reference)

パラメータテンプレートによる明示的なシンタックスの定義と登録の代わりに、テンプレートでは属性テンプレートを参照することによりこれらの二つの項目を指定してもよい。この節の使用は既に登録済みの属性の意味に、なんら影響を与えない。

9.5.1.4 振舞い (Behavior)

テンプレートは、パラメータの使用に対する全ての振舞いを定義する。

9.5.2 テンプレート構造 (Template structure)

<parameter-label> PARAMETER

CONTEXT context-type;

syntax-or-attribute-choice;

[BEHAVIOR <behavior-definition-label> [,<behavior-definition-label>]*;

]

[REGISTERED AS object-identifier];

supporting productions

Context-type -> context-keyword | ACTION-INFO | ACTION-REPLY |
EVENT-INFO | EVENT-REPLY | SPECIFIC-ERROR

Context-keyword -> type-reference.<identifier>

Syntax-or-attribute-choice -> WITH SYNTAX type-reference |
ATTRIBUTE <attribute-label>

9.5.3 サポートする定義 (Supporting definitions)

CONTEXT context-type

この節はパラメータが適用されるコンテキストを定義する。次のものが指定できる。

— context-keyword:

このオプションは、テンプレートの外部で定義されたコンテキストへの参照である。内容は、`type-reference` と `type-reference` で指定された管理 PDU のフィールド名を示す `identifier` から成る。それ故、他の文書で定義されたコンテキストへの参照として用いられる。例えば、CMIS イベント情報パラメータ (例、Rec.X733 | ISO/IEC 10164-4 の `problemData`) や CMIS アクション応答パラメータの特定フィールドにのみ、そのパラメータが適用されることを示すために用いられる。もしパラメータが特定のフィールド名 (例えば、イベント情報は、パラメータ ID と値のペアの集合として定義される) に対応しない場合は、以下の、より一般的なコンテキストの一つが指定できる。

— ACTION-INFO:

このオプションは、CMIS アクション情報パラメータで運ばれる情報の表現に適用されるように、パラメータのコンテキストを定義する。

— ACTION-REPLY:

このオプションは、CMIS アクション応答パラメータで運ばれる情報の表現に適用されるように、パラメータのコンテキストを定義する。

— EVENT-INFO:

このオプションは、CMIS イベント情報パラメータで運ばれる情報の表現に適用されるように、パラメータのコンテキストを定義する。

— EVENT-REPLY:

このオプションは、CMIS イベント応答パラメータで運ばれる情報の表現に適用されるように、パラメータのコンテキストを定義する。

— SPECIFIC-ERROR:

このオプションは、CMIS プロセス障害エラーの表現や生成に適用されるようにパラメータのコンテキストを定義する。このオプションが属性に適用されるパラメータと共に用いられる場合は、管理オブジェクトクラスの定義では、このエラーがある属性に対する属性値置換操作またはデフォルト値置換操作において発生した時、一つの置換要求内で参照される他の属性も変更されるか否かを指定しなければならない。

WITH SYNTAX type-reference

この節は、プロトコルで運ばれる際のパラメータの ASN.1 型を指定する。

ATTRIBUTE <attribute-label>

この節には、ATTRIBUTE テンプレートを指定する。指定した属性のシンタックスとオブジェクト識別子が、このパラメータのシンタックスとオブジェクト識別子として用いられる

BEHAVIOR <behavior-definition-label> [<behavior-definition-label>]*

この節は、パラメータに関する全ての振舞いとセマンテックスの指定を可能にする。もし ATTRIBUTE 節が用いられている場合、この節は属性の振舞いを変更しない。

REGISTERED AS <object-identifier>

object-identifier 値は、パラメータ定義のためのグローバルにユニークな識別子を提供する。この値は、パラメータを識別する必要がある時に管理プロトコルの中で用いられる。この節は、WITH SYNTAX 節が存在する場合にのみ必要となる。

9.6 ネームバインディングテンプレート (Name binding template)

9.6.1 概要 (Overview)

このテンプレートは、ある管理オブジェクトクラスの管理オブジェクトに対して、ネームバインディングによって別の名前構造を定義することを可能とする。ネームバインディングにより選択された属性はネーミング属性となり、その属性は、指定された管理オブジェクトクラスのインスタンス（下位オブジェクト）が、指定された管理オブジェクトクラス、または他のオブジェクトクラス（例えば、ディレクトリオブジェクトクラス）のインスタンス（上位オブジェクト）によって命名される時に用いられる。

ネームバインディングが用いられる場合、下位のオブジェクトにはネーミング属性が存在しなければならない。このネーミング属性は、下位オブジェクトの相対識別名 (RDN) を構成するために用いられる。RDN は、その属性型に割り当てられたオブジェクト識別子とその属性のインスタンスの値により構成される。下位オブジェクトの識別名は、上位オブジェクトの識別名に下位オブジェクトの RDN を付加することにより得られる。

ネームバインディングは、それらが参照するクラスの定義の一部ではない。ある下位の管理オブジェクトクラスは、一つ以上のネームバインディングを持ってもよい。ネームバインディングの集合は、上位オブジェクトとのネーミング関係の集合および下位オブジェクトが生成される元となる管理オブジェクトクラスの集合を定義する。

ネームバインディングはまた、指定された上位クラスの全てのサブクラス、および下位クラスの全てのサブクラスについても適用されるように定義できる。

注) 管理オブジェクトクラスに対するネームバインディングは、その管理オブジェクトクラス自身の指定の次に記述する。

9.6.2 テンプレート構造 (Template structure)

```
<name-binding-label> NAME BINDING
  SUBORDINATE OBJECT CLASS      <class-label> [AND SUBCLASSES];
  NAMED BY SUPERIOR OBJECT CLASS <class-label> [AND SUBCLASSES];
  WITH ATTRIBUTE                 <attribute-label>;
  [BEHAVIOR <behavior-definition-label> [, <behavior-definition-label>]*;
  ]
  [CREATE   [<create-modifier> [, <create-modifier>]] [<parameter-label>]*;
  ]
  [DELETE   [<delete-modifier>] [<parameter-label>]*;
  ]
REGISTERED AS <object-identifier>
supporting productions
create-modifier ->   WITH-REFERENCE-OBJECT|
                    WITH-AUTOMATIC-INSTANCE-NAMING
delete-modifier ->  ONLY-IF-NO-CONTAINED-OBJECTS
                    DELETES-CONTAINED-OBJECTS
```

9.6.3 サポートする定義 (Supporting definitions)

SUBORDINATE OBJECT CLASS <class-label> [AND SUBCLASSES]

ここでは、NAMED BY SUPERIOR OBJECT CLASS 節で定義されたオブジェクトクラスのインスタンスによって名付けられるインスタンスの管理オブジェクトクラスを定義する。この下位オブジェクトクラスのインスタンスの名前は、上位オブジェクトの識別名に下位オブジェクトの相対識別名を連結して構成される。AND SUBCLASSES が指定された場合は、そのネームバインディングは、指定された管理オブジェクトクラスの全てのサブクラスにも適用される。

NAMED BY SUPERIOR OBJECT CLASS <class-label> [AND SUBCLASSES]

ここでは、ある管理オブジェクトクラス、または (ディレクトリ・オブジェクトクラスのような)その他のクラスを定義する。このオブジェクトクラスのインスタンスが、SUBORDINATE OBJECT CLASS 節で定義された管理オブジェクトクラスのインスタンスを名前付ける。AND SUBCLASSES が指定された場合は、そのネームバインディングは、指定された管理オブジェクトクラスの全てのサブクラスにも適用される。

WITH ATTRIBUTE <attribute-label>

ここでは、SUBORDINATE OBJECT CLASS 節で定義された管理オブジェクトクラスのインスタンスの相対識別名を構成するために (このネームバインディングのコンテキスト中で) 用いられる属性を定義する。この属性の値は、Rec.X.720 | ISO/IEC 10165-1 で指定された制約に従って、単一値 (single-valued) のデータ型として表現されなければならない。もし、名前属性として使用できる適切な属性がない場合、管理オブジェクトの設計者は、GraphicString 型の名前属性をつけることを奨励されている。

BEHAVIOR <behavior-definition-label> [,<behavior-definition-label>]*

この節では、定義されたネームバインディングの結果として発生する全ての振舞いを指定する。
behavior-definition-label には、関係する振舞い定義を指定する。

注) この節は、ネームバインディングに特有の振舞いを記述するために用いられることを目的としたものである。管理オブジェクトクラスの全てのインスタンスに適用される振舞いは、管理オブジェクトクラスを定義するパッケージテンプレートから参照される振舞いの一部として、定義されなければならない。

CREATE [<create-modifier> [,<create-modifier>]] [<parameter-label>]*

この節は、このネームバインディングにおける SUBORDINATE OBJECT CLASS 節で参照される管理オブジェクトクラスの新しいインスタンスを、システム管理操作によって生成することを可能とする。
create-modifier の値は生成に対するオプションを指定し、以下の値が許される。

- WITH-REFERENCE-OBJECT.

生成時にデフォルト値のソースとして参照する管理オブジェクトが指定される。そして条件付きパッケージの選択にも、この管理オブジェクトが参照される。

- WITH-AUTOMATIC-INSTANCE-NAMING.

CREATE 要求で、新しい管理オブジェクトのインスタンス名を省略できる。

振舞い定義では、新しい管理オブジェクトに適用されるネームバインディングの選択があった場合、どのように振る舞うかを指定しなければならない。

管理オブジェクト生成時の属性の初期値のソース、およびそれに関連する優先ルールは Rec.720 | ISO/IEC 10165-1 に規定されている。

parameter-label には、生成操作に関するネームバインディング固有のエラーパラメータを指定する。これらは処理障害として報告される。エラーパラメータのシンタックスは、参照するテンプレートで定義される。

DELETE [<delete-modifier>] [<parameter-label>]*

この節は、このネームバインディングにおける SUBORDINATE OBJECT CLASS 節で参照される管理オブジェクトクラスのインスタンスを消去することを可能とする。delete-modifier には、管理オブジェクトが消去される時のその管理オブジェクトクラスの振舞いを指定し、次の値が許される。

- ONLY-IF-NO-CONTAINED-OBJECTS.

全ての包含される管理オブジェクトは、包含する管理オブジェクトの消去に先だって、管理操作によって明示的に消去されていなければならない。すなわち、消去要求は、包含される管理オブジェクトがまだ残っている場合、エラーとなる。

- DELETES-CONTAINED-OBJECTS.

消去要求が本修飾子を指定した管理オブジェクトに対して行われる場合、直接的または間接的に包含されたいずれかの管理オブジェクトに ONLY-IF-NO-CONTAINED-OBJECTS が指定され、しかも包含された管理オブジェクトをもつならば、その消去要求は失敗となる。それ以外の場合、消去要求は包含された管理オブジェクトも消去し、成功となる。

包含された管理オブジェクトの消去に関する振舞いを記述するその他の規則は、振舞い節で指定する。

注) DELETES-CONTAINED-OBJECT 修飾子は、包含している管理オブジェクトが存在するか否かにかかわらず、その上位の管理オブジェクトを消去することを許している。このため、どの修飾子が適当か疑問であるならば、ONLY-IF-NO-CONTAINED-OBJECTS 修飾子を用いるのが望ましい。

もし消去に関して、他の関係や管理オブジェクトクラス全般の条件に関する制約がある場合、それらは管理オブジェクトクラスの振舞いの一部として指定されなければならない。

parameter-label には、消去操作に関するネームバインディング固有のエラーパラメータを指定する。これらは処理障害として報告される。エラーパラメータのシンタックスは、参照するテンプレートで定義される。

REGISTERED AS <object-identifier>

object-identifier の値は、ネームバインディング定義の、グローバルにユニークな識別子を提供する。この値は、知識管理の目的でネームバインディングを識別するために用いられる。

9.7 属性テンプレート (Attribute template)

9.7.1 概要 (Overview)

このテンプレートは、個々の属性型を定義するために用いられる。これらの定義は、属性グループが必要な場合、属性グループテンプレートによって結合される。定義の主な要素を以下に示す。

9.7.1.1 導出 (Derivation)

属性型の定義では、他の属性型の定義を変更したり抑制したりすることができる。

9.7.1.2 属性シンタックス (Attribute syntax)

属性型の定義は、管理プロトコルで属性値を運ぶ時に用いられるシンタックスの定義を含んでいなければならない。この定義は、ASN.1 型定義を参照することにより可能となる。属性シンタックスの定義では、属性値が単一値 (single-valued) か多値 (set-valued) かのいずれの属性型であるかを指定する。基本型が SET OF ならば、属性は多値型であり、その他の場合は単一値型である。

9.7.1.3 値の照合 (Value matching)

属性型の定義には、その型のインスタンスの値がテストされる正当な方法 (等しいか否かや大きさ等のテスト) を含む。いくつかの属性型の値の照合には、その照合規則がどの様に適用されるかを、属性の振舞い定義の一部として指定する必要がある。属性定義に照合規則の指定がない場合は、値の照合は未定義であることを意味する。

9.7.1.4 振舞い (Behavior)

属性定義には、属性に固有の振舞いの定義を含むことができる。すなわち、どの管理オブジェクトクラスがその属性型のインスタンスを含むかにかかわらず、その属性型に適用される振舞いである。

9.7.1.5 属性識別子 (Attribute identifier)

オブジェクト識別子の値は、管理オブジェクトクラスの定義内に含まれる各属性にそれぞれ割り当てられなければならない。この値は管理プロトコルで属性を識別するために用いられる。

9.7.1.6 パラメータ (Parameters)

属性定義では、その属性型の管理操作に関する属性固有のエラーパラメータを指定できる。参照するテンプレートでは、これらを処理障害に対応づける。

9.7.2 テンプレート構造 (Template structure)

```
<attribute-label> ATTRIBUTE
  derived-or-with-syntax-choice;
  [MATCHES FOR    qualifier [,qualifier]*;
  ]
  [BEHAVIOR    <behavior-definition-label> [,<behavior-definition-label>]*;
  ]
  [PARAMETERS    <parameter-label> [,<parameter-label>]*;
  ]
[REGISTERED AS <object-identifier>];
supporting productions
qualifier ->
    EQUALITY | ORDERING | SUBSTRINGS |
    SET-COMPARISON | SET-INTERSECTION
derived-or-with-syntax-choice ->
    DERIVED FROM <attribute-label> |
    WITH ATTRIBUTE SYNTAX type-reference
```

9.7.3 サポートする定義 (Supporting definitions)

DERIVED FROM <attribute-label>

この要素が存在する場合、属性定義は、まず初めに `attribute-label` で参照した定義の全ての側面をとる。その中には、他の属性定義から結果として引き出されるもの全てが含まれる。この様な環境下での属性テンプレートの他の要素の存在の結果を解釈する規則を以下に示す。

- MATCHES FOR : 結果として生ずる照合規則の集合は、導出される全ての照合規則と、この節で指定される照合規則の論理和となる。
- BEHAVIOR : 導出される全ての振舞い定義を拡張する。
- REGISTERED AS : 他の定義から導出された全ての登録を置換する。

この導出メカニズムは、以下のことを可能とする。

- 既存の属性定義に基づく属性の定義
- 既存の属性定義への制約の付加

WITH ATTRIBUTE SYNTAX type-reference

この要素は、DERIVED FROM 節がない場合にのみ存在し、属性値のインスタンスがプロトコルによりどのように運ばれるかを示す ASN.1 データ型を指定する。

ASN.1 データ型は、その属性自身のデータ型をも定義する。シンタックスの基本型が単一型集合型 (set-of) ならば、属性は多値属性となる。その他の全ての ASN.1 データ型 (集合型(set)、順序型(sequence)、単一型順序型(sequence-of)も含め) は、単一値の属性型を定義する。

MATCHES FOR qualifier [,qualifier]*

この要素は、フィルタ操作の一部として属性値に適用される照合の型を指定する。属性の存在についての照合は暗黙的に全ての属性に対して許される。その他の型の照合については、もしこの節が存在しない場合、定義されず、従ってその属性に対しては許されない。以下のオプションがある。

- EQUALITY. 属性値は与えられた値と等しいか否かテストされる。
- ORDERING. 属性値と与えられた値との大小がテストされる。
- SUBSTRINGS. 属性値の中に与えられた部分文字列が存在するか否か、テストされる。
- SET-COMPARISON. 属性値と与えられた値とが、部分集合/全集合という関係にあるかテストされる。
- SET-INTERSECTION. 属性値と与えられた値との間に空でない共通集合が存在するか否かテストされる。

BEHAVIOR <behavior-definition-label> [,<behavior-definition-label>]*

この振舞い節は、本属性型に一般的な振舞いを定義する。振舞い定義は、選択された照合規則が属性定義にどのように適用されるかを定義するために必要な付加的な指定を含まなければならない。管理オブジェクトクラスに固有な振舞いはパッケージテンプレートの振舞い節で定義する。

PARAMETERS <parameter-label> [,<parameter-label>]*

parameter-label は、処理障害の定義のために、パラメータを属性型の振舞いと関連づけることを可能とする。例えば、ある属性型はある条件下で " 制約違反 " エラーを表示する。そのようなエラーに関する情報を与えるパラメータは、CONTEXT SPECIFIC-ERROR を持つパラメータテンプレートをを用いて指定される。このパラメータは、属性テンプレートから参照できる。

REGISTERED AS <object-identifier>

object-identifier の値は、属性定義のためのグローバルにユニークな識別子を提供する。これは、DERIVED FROM, WITH ATTRIBUTE SYNTAX, MATCHES FOR, BEHAVIOR の各節によって直接、または間接に参照される全ての要素を含む。この値は属性型を識別する必要がある時に、管理プロトコルの中で用いられる。この節が省略された場合は、その属性定義は管理オブジェクトクラスの定義の中で参照されてはならない。属性定義が REGISTERED AS 節を含む既存の属性定義から引き出される場合、既存の定義に割り当てられている OBJECT IDENTIFIER は引き出された定義の正しい識別子とはならない。従って、REGISTERED AS 節は、それが管理オブジェクトクラスの定義内で参照されるならば、引き出された定義の中に含まれなければならない。

9.8 属性グループテンプレート (Attribute group template)

9.8.1 概要 (Overview)

このテンプレートは、属性グループを定義する為のものである。グループ化は、グループを構成する属性の集合に対して操作を行いたいような状況に対応するために行う。属性グループに対して属性値取得操作およびデフォルト値置換操作を適用する場合、管理オブジェクトクラスの振舞い定義で、その方法を定義する。グループの各メンバは、属性型が単一値 (single-valued) 型または多値 (set-valued) 型と定義されていなければならない。

属性グループテンプレートでは、グループを構成する属性の最小セットと、そのグループの名前付けに使用されるオブジェクト識別子の値を定義する。属性グループを参照する個々の管理オブジェクトクラス定義では、グループが固定メンバ (fixed membership) と定義されていない限り、グループに新しい属性を追加することによりグループのメンバを拡張することができる。この拡張は、その拡張が定義されているオブジェクトクラスのインスタンスに対してのみ適用される。属性グループテンプレートで指定される属性では、そのグループを参照する全ての管理オブジェクトクラス定義におけるグループの最小構成要素を定義する。

管理オブジェクトクラス定義の中に拡張 (extensible) 属性グループが存在する場合、そのグループに対し定義されたすべての属性 (属性グループテンプレートの中で定義されているか、管理オブジェクトクラス定義により属性グループに追加されたもの) は、そのグループを参照するパッケージの中か、管理オブジェクトクラスの必須パッケージの一つの中のいずれかに存在しなければならない。

管理オブジェクトクラスの中に固定 (fixed) 属性グループが存在する場合、そのグループに定義された全ての属性は、そのグループを参照するパッケージの中に存在しなければならない。

9.8.2 テンプレート構造 (Template structure)

```
<group-label> ATTRIBUTE GROUP
  [GROUP ELEMENTS <attribute-label> [,<attribute-label>]*;
  ]
  [FIXED ;
  ]
  [DESCRIPTION delimited-string ;
  ]
REGISTERED AS object-identifier ;
```

9.8.3 サポートする定義 (Supporting definitions)

```
GROUP ELEMENTS <attribute-label> [,<attribute-label>]*
```

この節では、属性グループの全てのインスタンスの中に存在する個々の属性を識別する attribute-label を定義する。個々の属性は、ATTRIBUTE テンプレートを使用して定義される。属性グループに対して属性値取得操作およびデフォルト値置換操作を適用する場合、管理オブジェクトクラスの振舞い定義で、その方法を定義する。

注：このことは、属性が個々に取り扱われる場合にはその属性に対し適用されない属性グループ固有の振舞い仕様が存在することを意味するものではない。

グループ内の全ての属性は、そのグループを参照する管理オブジェクトクラス定義のメンバでなければならない。すなわち、ある管理オブジェクトクラスにおけるグループのメンバである個々の属性は、その管理オブジェクトクラスが参照する一つ以上のパッケージの中の **Attributes** 節で参照されていないとなければならない。

FIXED

この節は、属性グループが固定メンバ (**fixed membership**) と定義されていることを示す。

DESCRIPTION delimited-string

この節は、グループ化の意味を記述するものである。たとえば、“管理オブジェクトの全ての状態属性のグループである”等。**DESCRIPTION** の記述に使用する文字セットの制限は無い。また、**DESCRIPTION** の中の構造についても定義は無い。

この節は、グループおよびそのメンバの振舞いを定義する目的で使用してはならない。

REGISTERED AS object-identifier ;

object-identifier の値は、属性グループ定義のグローバルでユニークな識別子となる。この値は、管理プロトコルにおいて、属性グループを識別する必要がある場合に使用される。管理オブジェクトのコンテキスト中の **object-identifier** で識別される属性グループは、属性グループテンプレートの中で定義された全ての属性と、拡張可能属性グループの場合、管理オブジェクトのインスタンス生成時に適用される管理オブジェクトクラステンプレートの要素の定義の結果としてグループに追加される全ての属性を含む。

注：属性グループテンプレートでは、常にそのグループのメンバとなる属性の集合（空集合の場合もある）を定義する。拡張可能属性グループの場合、この集合は、特定の管理オブジェクトクラス定義のために集合を拡張する目的で、パッケージテンプレートの **Attribute Groups** 節を使用して拡張できる。この技法は、次の様な場合に属性グループを定義することにより適用できる。すなわち、グループのメンバが（状態属性のような）共通の意味を持つが、ある管理オブジェクトクラスに存在するその意味の型を持つ属性の数がインスタンス生成時にのみ決定される場合、あるいは、複数の管理オブジェクトクラスで同じ意味を持つ属性の異なるグループ化が要求される場合である。一般的に、拡張可能属性グループの構成は、管理オブジェクトの生成時にのみ決定できる。その時点で、どのパッケージが、即ち、どの属性が生成されたかが認識できるためである。

9.9 振舞いテンプレート (Behavior template)

9.9.1 概要 (Overview)

このテンプレートは、管理オブジェクトクラス、ネームバインディング、パラメータと属性、アクションおよび通知型の振舞いを定義するために使用する。振舞いテンプレートは、定義の拡張を行うためのものであるが、振舞い記述では、以前に定義された情報の意味を変更してはならない。未定義の情報がある場合、振舞い定義で何が未定義であることを明示しなければならない。

注

1. 振舞いテンプレートは、他のテンプレートで完全に記述されていない意味を示すために使用すべきである。特に、定義者は、意味を持つラベルにたよってはならない。
2. 振舞い文は、管理オブジェクトクラス定義に含まれている管理オブジェクトの表現方法で記述すべきである。

9.9.2 テンプレート構造 (Template structure)

```
<behavior-definition-label> BEHAVIOR  
    DEFINED AS delimited-string ;
```

9.9.3 サポートする定義 (Supporting definitions)

```
DEFINED AS delimited-string ;
```

`delimited-string` では、管理オブジェクトクラスや対応するネームバインディング、パラメータ、属性、アクションまたは通知の振舞いを定義する。この定義は、通常の言語または形式記述技法を用いて記述する。テキストは、文書または勧告／標準の章、節からのテキストの引用でもよい。`delimited-string` の記述に使用する文字セットの制限は無い。また、テキストの構造についての定義もない。

9.10 アクションテンプレート (Action template)

9.10.1 概要 (Overview)

このテンプレートは、個々のアクション型の振舞いと構文 (syntax) を定義するために使用する。このテンプレートで定義されたアクション型は、**Rec.X.710 | ISO/IEC 9595** で定義されている **M-ACTION** サービスで使用される。この定義の主な特徴を以下に示す。

9.10.1.1 振舞い (Behavior)

アクション型の定義では、操作される管理オブジェクトクラスに与える影響の側面から、アクションの機能を指定する。一つ以上の管理オブジェクトクラスに対しアクションが適用される場合、振舞いの記述は、全ての管理オブジェクトクラスに共通な振舞いの機能に限定しなければならない。それらのアクションに関係する振舞いで、特定の管理オブジェクトに固有なものは、その管理オブジェクトクラスの定義の一部として記述する。

9.10.1.2 操作モード (Mode of operation)

アクション型の定義では、アクションが常に確認型であるか、マネージャの選択により確認型または非確認型となるかを、指定する。

9.10.1.3 抽象構文 (Abstract syntax)

アクション型の定義では、Rec.X.710 | ISO/IEC 9595 で規定されている M-ACTION サービスのアクション情報 (action information) パラメータおよびアクション応答 (action reply) パラメータを運ぶために使用される構文を指定する。この構文は、ASN.1 のデータ型により定義される。

注：

1. アクションの引数の将来的な拡張を意図的に禁止するのであれば、アクションの 情報/応答構文は拡張可能な様に定義しておくべきである。これは、(Rec.X.721 | ISO/IEC 10165-2 で定義されている様に) ASN.1 型の SET OF MngmntExtension をオプションフィールドとして指定することにより可能となる。
2. 情報/応答構文のために選択される基本データ型は、順序型が望ましい。

9.10.1.4 アクション識別子 (Action identifiers)

アクション型定義に関連するオブジェクト識別子の値は、管理プロトコル上でアクション型を識別するために使用される。

9.10.1.5 パラメータ (Parameters)

アクション型の定義では、そのアクション型に関するアクション情報パラメータやアクション応答パラメータまたは固有誤りパラメータを指定する。

9.10.2 テンプレート構造 (Template structure)

```
<action-label> ACTION
  [BEHAVIOR <behavior-definition-label> [,<behavior-definition-label>]*;
  ]
  [MODE CONFIRMED ;
  ]
  [PARAMETERS <parameter-label> [,<parameter-label>]*;
  ]
  [WITH INFORMATION SYNTAX type-reference ;
  ]
  [WITH REPLY SYNTAX type-reference ;
  ]
REGISTERED AS object-identifier ;
```

9.10.3 サポートする定義 (Supporting definitions)

```
BEHAVIOR <behavior-definition-label> [,<behavior-definition-label>]*
```

この節では、アクションの振舞い、アクションに指定されるべきパラメータ、アクションが引き起こす結果とそれらの意味を定義する。behavior-definition-labels では、振舞いテンプレートで定義された振舞い記述を参照する。

MODE CONFIRMED

指定された場合、アクションは確認型として機能する。省略された場合、アクションはマネージャの判断により、確認型または非確認型になる。

PARAMETERS <parameter-label> [,<parameter-label>]*

parameter-label では、そのアクション型に関するアクション情報パラメータやアクション応答パラメータまたは処理障害を指定する。A.7 の例を参照のこと。

WITH INFORMATION SYNTAX type-reference

type-reference は、管理プロトコル上を運ばれるアクション情報パラメータの構造を記述する ASN.1 のデータ型を示す。省略された場合、アクション要求に関する固有の情報はない。

WITH REPLY SYNTAX type-reference

type-reference は、管理プロトコル上を運ばれるアクション応答パラメータの構造を記述する ASN.1 のデータ型を示す。省略された場合、アクション応答に関する固有の情報はない。

REGISTERED AS object-identifier ;

object-identifier の値は、アクション型定義のグローバルでユニークな識別子となる。この値は、管理プロトコルで、アクション型の識別が必要な場合に使用される。

9.11 通知テンプレート (Notification template)

9.11.1 概要 (Overview)

このテンプレートは、個々の通知型の振舞いと構文を定義するために使用する。このテンプレートで定義された通知型は、Rec.X.710 | ISO/IEC 9595 で規定されている M-EVENTREPORT サービスによる事象報告で使用される。この定義の主な特徴を以下に示す。

9.11.1.1 振舞い (Behavior)

通知型の定義では、その型の通知が生成される状況を指定する。

9.11.1.2 抽象構文 (Abstract syntax)

通知型の定義では、Rec.X.710 | ISO/IEC 9595 で規定されている M-EVENT REPORT サービスの事象情報 (event information) パラメータおよび事象応答 (event reply) パラメータを運ぶために使用される構文を指定する。このテンプレートでは、構文中のフィールドに属性値を割り当てることもできる。

注：

1. 通知の引数の将来的な拡張を意図的に禁止するのでなければ、通知の情報／応答構文は拡張可能な様に定義しておくべきである。これは、(Rec.X.721 | ISO/IEC 10165-2 で定義されている様に) ASN.1 型 SET OF MngmntExtension をオプションフィールドとして指定することにより可能となる。

2. 情報/応答構文のために選択される基本データ型は、順序型が望ましい。

9.11.1.3 通知の名前付け (Notification naming)

通知定義に関連するオブジェクト識別子の値は、管理プロトコル上で事象 (Event) 型を識別するために使用される。

9.11.1.4 パラメータ (Parameters)

通知型の定義では、その通知型に関する事象情報パラメータ、事象応答パラメータまたは固有誤りパラメータを指定する。

9.11.2 テンプレート構造 (Template structure)

```
<notification-label> NOTIFICATION
    [BEHAVIOR <behavior-definition-label> [,<behavior-definition-label>]*;
    ]
    [PARAMETERS <parameter-label> [,<parameter-label>]*;
    ]
    [WITH INFORMATION SYNTAX type-reference
        [AND ATTRIBUTE IDS <field-name><attribute-label>
            [,<field-name><attribute-label>]*
        ];
    ]
    [WITH REPLY SYNTAX type-reference ;
    ]
REGISTERED AS object-identifier ;
```

9.11.3 サポートする定義 (Supporting definitions)

```
BEHAVIOR <behavior-definition-label> [,<behavior-definition-label>]*
```

この節では、通知の振舞い、通知に指定されるべきデータ、通知が引き起こす結果とそれらの意味を定義する。behavior-definition-label では、振舞いテンプレートで定義された振舞い記述を参照する。

PARAMETERS <parameter-label> [,<parameter-label>]*

parameter-label では、その通知型に関する事象情報パラメータ、事象応答パラメータまたは処理障害を指定する。A.3 の例を参照のこと。

WITH INFORMATION SYNTAX type-reference

[AND ATTRIBUTE IDS <field-name><attribute-label>
[,<field-name><attribute-label>]*

この節では、管理プロトコル上を運ばれる通知情報の構造を記述する ASN.1 のデータ型を示す。また、抽象構文中で名付けられたフィールドに属性識別子を関係付けることができる。省略された場合、通知要求に関する固有な情報はない。AND ATTRIBUTE IDS オプションが指定された場合、field-name は、この節で指定した type-reference で参照される抽象構文の中で定義されているラベルでなければならない。field-name でラベル付けられたデータ型が、attribute-label で参照される属性の値を運ぶために使用される。属性の ASN.1 データ型は、field-name で参照されるデータ型と同じでなければならない。

SET OF 型または SEQUENCE OF 型の中のラベル、あるいは、それらの要素のラベルは、field-name には使用できない。なぜならば、そのような繰り返し構文の中のラベルは、明瞭にあるデータ型の単一インスタンスを参照できないからである。同様に、CHOICE、SET または SEQUENCE 型の構成要素のラベル、あるいは、それらの構成要素の定義の中のラベルは、ラベルをつけられた構成要素が型定義中に複数回現れる場合は、field-name に使用できない。

WITH REPLY SYNTAX type-reference

この節は、管理プロトコル上を運ばれる通知応答の構造を記述する ASN.1 のデータ型を示す。省略された場合、通知応答に関する固有な情報はない。

応答構文は、通知が CMIS M-EVENT-REPORT サービスの確認モードを使用して送られた場合に使用される。事象応答は、管理オブジェクトに対しては返されない。通知を確認モードで送るか非確認モードで送るかは、エージェント側で、マネージャとの間でとりきめたポリシーに基づいて決定される。通知定義では WITH REPLY SYNTAX が省略されているが、通知が確認モードで送られた場合、確認には応答情報は含まれない。(CMIS サービスが、M-EVENT-REPORT 確認の事象応答パラメータを捨てる)

REGISTERED AS object-identifier ;

object-identifier の値は、通知型定義のグローバルでユニークな識別子となる。この値は、管理プロトコルで、通知型の識別が必要な場合に使用される。

付録 A

ガイドラインの使用例

(この付録は、この仕様の本体を構成するものではない。)

この付録に示した例は、テンプレートの記法の使用例を示すものであり、実装時に有効となる定義を示したものではない。特に、これらの例につけた振舞い定義は、必然的に、意図的なものとなっている。管理オブジェクトクラス定義の開発を実際に行う場合に有用な記法の使用例は、Rec.X.721 | ISO/IEC 10165-2 に示されている。

A. 1 管理オブジェクトクラス定義 (Managed object class definition)

exampleObjectClass MANAGED OBJECT CLASS

DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992".top ;

CHARACTERIZED BY examplePackage2 ;

CONDITIONAL PACKAGES

examplepackage1 PACKAGE

ACTIONS qosResetAction ;

NOTIFICATIONS communicationError ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) package(4) examplepack1(0)} ;

PRESENT IF !conformance class 2 of underlying resource implemented as described in ISO/IEC XXXX! ;

訳注：!ISO/IEC XXXX の記述に従い導入されている下位の資源の適合性クラス 2!

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) managedObjectClass(3) exampleclass(0)} ;

注：このテンプレートでは、条件付きパッケージの記述のために、インライン記述オプションを使用している。

A. 2 ネームバインディング定義 (Name binding definition)

exampleNameBinding NAME BINDING

SUBORDINATE OBJECT CLASS exampleObjectClass ;

NAMED BY

SUPERIOR OBJECT CLASS "Rec. X.721 | ISO/IEC 10165-2 : 1992".system ;

WITH ATTRIBUTE objectName ;

BEHAVIOR

containmentBehavior BEHAVIOR

DEFINED AS !A maximum of 3 instances of exampleObjectClass may be contained in any instance of "Rec. X.721 | ISO/IEC 10165-2".system.!

訳注：!"Rec. X.721 | ISO/IEC 10165-2".system の任意のインスタンス中には、最大3つの exampleObjectClass のインスタンス含まれる。!

;
;

```
CREATE WITH-AUTOMATIC- INSTANCE-NAMING createErrorParameter ;
DELETE DELETES-CONTAINED-OBJECTS ;
```

```
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) nameBinding(6) examplenb(0)} ;
```

注：このテンプレートでは、振舞いの記述のために、インライン記述オプションを使用している。

A. 3 パラメータ定義 (Parameter definition)

```
pDUHeader PARAMETER
```

```
CONTEXT EVENT-INFO;
```

```
WITH SYNTAX ParameterModule.PDUString;
```

```
BEHAVIOR
```

```
    pDUHeaderBehavior BEHAVIOR
```

```
        DEFINED AS !PDU header. Carried in the CMIP eventInfo field.!
```

訳注：!CMIP の eventInfo フィールドで運ばれる PDU ヘッダ!

```
    ;
```

```
    ;
```

```
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) parameter(5) pduheaderparam(0)};
```

```
createErrorparameter PARAMETER
```

```
CONTEXT SPECIFIC-ERROR ;
```

```
WITH SYNTAX parameterModule.ErrorInfo1 ;
```

```
BEHAVIOR
```

```
    createErrorBehavior BEHAVIOR
```

```
        DEFINED AS !If the maximum number of instances of exampleObjectClass exist within the containing managed object, attempts to create additional instances will result in the return of a CMIP Processing Failure error where the SpecificErrorInfo field is of the form:
```

```
SpecificErrorInfo SEQUENCE {
    errorid OBJECT IDENTIFIER,
    errorinfo ANY DEFINED BY errorid }
```

The OBJECT IDENTIFIER carried in errorid shall be the value under which this parameter definition is registered. The type carried in errorinfo shall be the type identified by the WITH SYNTAX construct of this parameter definition. The value carried by this type indicates the number of instances of this managed object class that currently exist in the containing managed object.!

訳注：!包含管理オブジェクト内に、exampleObjectClass のインスタンスが最大数存在する場合、新規のインスタンスの生成操作は、CMIP の処理障害 (Processing Failure) エラーを返す結果となる。この場合、SpecificErrorInfo フィールドはつぎの形式となる：

```

SpecificErrorInfo      SEQUENCE {
    errorid             OBJECT IDENTIFIER,
    errorinfo           ANY DEFINED BY errorid }

```

errorid で運ばれるオブジェクト識別子は、このパラメータ定義が登録された値となる。errorinfo で運ばれる型は、このパラメータ定義の WITH SYNTAX で指定された型である。この型で運ばれる値は、包含管理オブジェクト中に存在するこの管理オブジェクトクラスのインスタンスの数である。!

```

;
;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) parameter(5) createrror(1)} ;

```

serviceproviderErrorResponseReason PARAMETER

CONTEXT ACTION-REPLY;

WITH SYNTAX parameterModule.ServiceProviderErrorResponseReason;

BEHAVIOR

serviceproviderErrorResponseReasonBehavior BEHAVIOR

DEFINED AS !Returned in the responseParameters field of the CMIP actionReplyInfo if responseCode has the value serviceProviderErrorResponse.!

訳注：!responseCode が値 serviceProviderErrorResponse を持つ場合、CMIP の actionReplyInfo の responseParameters フィールドで返される。

```

;
;
REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) parameter(S) sperrorrsp(2)};

```

注：これらのテンプレートでは、振舞いの記述のために、インライン記述オプションを使用している。

A. 4 パッケージ定義 (Package definition)

examplepackage2 PACKAGE

```

BEHAVIOR      exampleClassBehavior ;
ATTRIBUTES   objectName      GET ,
              qOS-Error-Cause GET ,
              qOS-Error-Counter
              PERMITTED VALUES AttributeModule.QOSCounterRange
              REQUIRED VALUES  AttributeModule.QOSCounterRange
              GET ;
ATTRIBUTE GROUPS qQS-Group ;
NOTIFICATIONS  protocolError;

```

```

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) package(4) examnlepack2(1)} ;

```

注：このテンプレートでは、条件付きパッケージを使用していないため、REGISTERED AS 節は必ずしも必要ではない。しかし、後でこのパッケージを条件付きパッケージとして使用する必要が生じた時に追加するよりは、定義の時点で登録しておいた方が楽である。

A. 5 属性定義 (Attribute definition)

objectName ATTRIBUTE

WITH ATTRIBUTE SYNTAX AttributeModule.ObjectName ;
MATCHES FOR EQUALITY ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) attribute(7) objectname(0)} ;

qOS-Error-Cause ATTRIBUTE

WITH ATTRIBUTE SYNTAX AttributeModule.QOSErrorCause ;
MATCHES FOR EQUALITY ;
BEHAVIOR qOSErrorBehavior ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) attribute(7) qoscause(1)} ;

qOS-Error-Counter ATTRIBUTE

WITH ATTRIBUTE SYNTAX AttributeModule.QOSErrorCounter ;
MATCHES FOR EQUALITY, ORDERING ;
BEHAVIOR qOSCounterBehavior ;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) attribute(7) qoscount(2)} ;

A. 6 属性グループ定義 (Attribute group definition)

qOS-Group ATTRIBUTE GROUP

GROUP ELEMENTS qOS-Error-Cause, qOS-Error-Counter ;
DESCRIPTION !Attribute group that includes all QOS-related attributes
in a managed object class! ;

訳注：!管理オブジェクトクラス中の全ての QOS 関連の属性を含む属性グループである。!

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) attributeGroup(8) qosgroup(0)} ;

A. 7 アクション定義 (Action definition)

qOSResetAction ACTION

BEHAVIOR

reset BEHAVIOR

DEFINED AS !<Definition of the reset behavior and its effect
on the operation of the managed object, etc..>!

訳注：!管理オブジェクトの操作におけるリセットの振舞いとその影響等の定義。!

;
;

MODE CONFIRMED;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) action(9) reset(0)};

注：このアクション定義では、アクションの振舞いテンプレートのためにインライン記述オプションを使用している。要求 (invoke) および応答 (reply) のための抽象構文は定義されていない。

activate ACTION

BEHAVIOR

activateBehavior BEHAVIOR

DEFINED AS !Enables the managed object for operation. If the action succeeds, the value successResponse is returned in the responseCode parameter of the CMIP actionReplyInfo. If the action fails because of a problem with the underlying service provider, responseCode is set to the value serviceProviderErrorResponse and the parameter serviceProviderErrorResponseReason returned to indicate the cause of the problem.!

訳注：!管理オブジェクトを操作可能とする。処理が成功した場合、CMIP の actionReplyInfo の responseCode パラメータで値 successResponse が返される。下位のサービス提供者の問題で処理が失敗した場合、responseCode には値 serviceProviderErrorResponse が、パラメータ serviceProviderErrorResponseReason には、問題の原因が返される。!

;
;

MODE CONFIRMED;

PARAMETERS serviceProviderErrorResponseReason;

WITH REPLY SYNTAX ActionModule.ActivateReply;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) action(9) activate(1)};

A. 8 通知定義 (Notification definition)

CommunicationError NOTIFICATION

BEHAVIOR communicationErrorBehavior;

WITH INFORMATION SYNTAX NotificationModule.ErrorInfo;

WITH REPLY SYNTAX NotificationModule.ErrorResult;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) notification(10) commerror(0)} ;

protocolError NOTIFICATION

BEHAVIOR

protocotErrorBehavior BEHAVIOR

DEFINED AS !Generated when a protocol entity receives a PDU which is invalid or contains a protocol error. The notification includes the header of the received PDU.!

訳注：!プロトコルエンティティが、不正なPDUまたはプロトコルエラーのPDUを受け取った場合に生成される。通知には受信したPDUのヘッダを含む。!

;
;

PARAMETERS pduHeader;

WITH INFORMATION SYNTAX NotificationSyntax.ProtocolError;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) notification(10) protoerror(1)};

注：このテンプレートでは、振舞いの記述のためにインライン記述オプションを使用している。

A. 9 振舞い定義 (Behavior definition)

qOSCounterBehavior BEHAVIOR

DEFINED AS !The QOS Error Counter attribute is a wraparound counter that is incremented by one for every occurrence of a QOS Error. Its value is a positive integer, whose range is specified in any package that references this definition. when the counter reaches its maximum value, the next increment causes its value to return to zero.!

訳注：!QOS エラーカウンタは、一巡するカウンタで、QOS エラーの発生毎に1ずつ増加する。値は正の整数で、その範囲は、この定義を参照するパッケージで指定される。カウンタが最大値に達した場合、次の増加時に0に戻る。!

qOSErrorBehavior BEHAVIOR

DEFINED AS !The QOS Error Cause attribute indicates the reason for a failure in quality of service associated with the managed object.

NOTE - The relationship between the permitted attribute values and the operation of the managed object itself are defined by the behavior definitions associated with the managed object class definition. !;

訳注：!QOS エラー原因属性は、その管理オブジェクトに関するサービス品質の障害の原因を示す。

注：許容される属性値とその管理オブジェクト自信の操作との関係は、その管理オブジェクトクラス定義の振舞い定義で指定される。!

communicationErrorBehavior BEHAVIOR

DEFINED AS !The CommunicationError notification is generated by the managed object class when a communication error is detected by the managed object. The notification may contain any combination of the parameters Probable Cause, Severity, Trend Indication, Backed Up Status, Diagnostic Info, Proposed Repair Action, Threshold Info, State Change and Other Info.

NOTE - The precise definition of what constitutes a communication error and the parameter values that apply is managed object class specific. In a practical example, this Behavior definition could, for example, refer to pieces of specification in a base standard in order to specify the behavior. !;

訳注 : !CommunicationError 通知は、管理オブジェクトが通信エラーを検知した場合、管理オブジェクトクラスが生成する。通知には、次のパラメータの任意の組み合わせが含まれる : Probable Cause, Severity, Trend Indication, Backed Up Status, Diagnostic Info, Proposed Repair Action, Threshold Info, State Change および Other Info.

注 : 通信エラーと適用されるパラメータの値の細かな定義は、管理オブジェクトクラス毎に異なる。現実的な例としては、振舞い定義において、振舞いを定義するために基本標準のいくつかの指定を参照することがあげられる。!

exampleClassBehavior BEHAVIOR

DEFINED AS !<Description of managed object class behavior, including:
- How its attributes attain particular values & what they mean,
- What circumstances cause notifications to be generated,
- Etc. >!

訳注 : !管理オブジェクトクラスの振舞いの記述で、次のものを含む :
- その属性がどのように特別な値をとるか、またその意味は、
- どのような状況で通知は生成されるか、
等。!

A. 10 ASN.1 モジュール (ASN.1 modules)

```
AttributeModule {joint-iso-ccitt ms(9) smi(3) part4(4) asn1Module(2) attributes(0)}  
DEFINITIONS ::= BEGIN
```

```
ObjectName ::= GraphicString
```

```
QOSErrorCause ::= INTEGER {  
    responseTimeExcessive (0),  
    queueSizeExceeded (1),  
    bandwidthReduced (2),  
    retransmissionRateExcessive (3) }
```

```
QOSErrorCounter ::= INTEGER
```

```
QOSErrorCounter ::= QOSErrorCounter {0..4294967296} -- Range is 32 bits
```

```
END
```

```
NotificationModule {joint-iso-ccitt ms(9) smi(3) part4(4) asn1Module(2) notifications(1)}  
DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
ProbableCause, PerceivedSeverity, TrendIndication, BackedUpstatus,  
ProposedRepairActions, ThresholdInfo, MngmntExtension
```

```
FROM Attribute.ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2)  
    asn1Module(2) 1};
```

```
ErrorInfo ::= SET {  
  
    [0] ProbableCause OPTIONAL,  
    [1] PerceivedSeverity OPTIONAL,  
    [2] TrendIndication OPTIONAL,  
    [3] BackedUpStatus OPTIONAL,  
    [4] ProposedRepairActions OPTIONAL,  
    [5] ThresholdInfo OPTIONAL,  
    [6] OtherInfo OPTIONAL }  
}
```

```
ErrorResult ::= NULL
```

```
OtherInfo ::= SET OF MngmntExtension
```

ProtocolError ::= SET OF MngmntExtension

END

ActionModule {joint-iso-ccitt ms(9) smi(3) part4(4) asn1Module(2) actions(2)}

DEFINITIONS ::= BEGIN

IMPORTS

Operationalstate, MngmntExtension

FROM Attribute.ASN1Module {joint-iso-ccitt ms(9)smi(3) part2(2) asn1Module(2) 1};

ActivateReply ::= SEQUENCE {

Operationalstatus [0] OperationalState,

ResponseCcode [1] INTEGER {successResponse (0),
serviceProviderErrorResponse (1) },

responseParams [2] SET OF MngmntExtension OPTIONAL }

END

ParameterModule {joint-iso-ccitt ms(9) smi(3) part4(4) asn1module(2) parameters(3)}

DEFINITIONS ::= BEGIN

ErrorInfo1 ::= INTEGER

ServiceProviderErrorResponseReason ::= ENUMERATED {

insufficientResources (0),

providerDoesNotExist (1),

providerNotAvailable (2),

requiredServiceNotAvailable (3) }

PDUString ::= OCTETSTRING

END