



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

**CCITT**

**X.722**

COMITÉ CONSULTATIF  
INTERNATIONAL  
TÉLÉGRAPHIQUE ET TÉLÉPHONIQUE

(01/92)

**RÉSEAUX DE COMMUNICATIONS DE DONNÉES**

---

**TECHNOLOGIES DE L'INFORMATION –  
INTERCONNEXION DE SYSTÈMES  
OUVERTS – STRUCTURE DES  
INFORMATIONS DE GESTION: DIRECTIVES  
POUR LA DÉFINITION DES OBJETS GÉRÉS**



**Recommandation X.722**

---

## Avant-propos

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. Le CCITT (Comité consultatif international télégraphique et téléphonique) est un organe permanent de l'UIT. Au sein du CCITT, qui est l'entité qui établit les normes mondiales (Recommandations) sur les télécommunications, participent quelque 166 pays membres, 68 exploitations privées reconnues, 163 organisations scientifiques et industrielles et 39 organisations internationales.

L'approbation des Recommandations par les membres du CCITT s'effectue selon la procédure définie dans la Résolution n° 2 du CCITT (Melbourne, 1988). De plus, l'Assemblée plénière du CCITT, qui se réunit tous les quatre ans, approuve les Recommandations qui lui sont soumises et établit le programme d'études pour la période suivante.

Dans certains secteurs de la technologie de l'information qui correspondent à la sphère de compétence du CCITT, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI. Le texte de la Recommandation X.722 du CCITT a été approuvé le 17 janvier 1992. Son texte est publié, sous forme identique, comme Norme internationale ISO/CEI 10165-4.

---

### NOTE DU CCITT

Dans cette Recommandation, l'expression «Administration» est utilisée pour désigner de façon abrégée aussi bien une Administration de télécommunications qu'une exploitation privée reconnue.

© UIT 1993

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'éditeur.

## SOMMAIRE

	<i>Page</i>
1	Domaine d'application..... 1
2	Références normatives..... 2
2.1	Recommandations du CCITT   Normes internationales identiques ..... 2
2.2	Paires de Recommandations du CCITT   Normes internationales équivalentes par leur contenu technique ..... 2
3	Définitions ..... 3
3.1	Définitions relatives au modèle de référence de base ..... 3
3.2	Définitions relatives à la dénomination et à l'adressage ..... 3
3.3	Définitions relatives au cadre général de gestion ..... 3
3.4	Définitions relatives à l'aperçu général de la gestion des systèmes ..... 3
3.5	Définitions relatives au modèle d'information de gestion ..... 4
3.6	Définitions relatives au service commun d'information de gestion (CMIS) ..... 4
3.7	Définitions relatives à l'ASN.1 ..... 5
3.8	Définitions complémentaires..... 5
4	Abréviations ..... 5
5	Conventions..... 6
6	Questions générales ..... 6
6.1	Intégrité des relations ..... 6
6.2	Caractéristiques héritées..... 6
6.3	Proposition d'options ..... 6
6.4	Enregistrement ..... 7
6.5	Conformité ..... 13
6.6	Complexité des définitions des objets gérés..... 13
6.7	Création et suppression d'objets gérés..... 13
7	Principes généraux pour la définition des objets gérés..... 14
7.1	Éléments communs ..... 14
7.2	Que gérer? ..... 14
7.3	Structuration ..... 14
7.4	Objets gérés..... 15
7.5	Attributs ..... 16
7.6	Relations entre valeurs d'attribut ..... 16
7.7	Modélisation des SAP ..... 17
7.8	Statistiques ..... 17
7.9	Compteurs ..... 18
7.10	Temporisateurs ..... 18
7.11	Mise à jour des attributs ..... 18
7.12	Précision des attributs ..... 18
7.13	Identification des objets gérés ..... 19
7.14	Notifications ..... 19
7.15	Utilisation des opérations ..... 19

8	Outils de notation pour la définition des objets gérés .....	20
8.1	Vue d'ensemble des outils de notation .....	20
8.2	Conventions utilisées dans les définitions de modèle .....	20
8.3	Modèle de classe d'objets gérés.....	24
8.4	Modèle d'ensemble.....	26
8.5	Modèle de paramètre.....	29
8.6	Modèle de corrélation des noms.....	32
8.7	Modèle d'attribut .....	35
8.8	Modèle de groupe d'attributs .....	37
8.9	Modèle de comportement.....	38
8.10	Modèle d'action .....	39
8.11	Modèle de notification .....	41
ANNEXE A	– Exemples d'utilisation des directives.....	43
A.1	Définition de classe d'objets gérés .....	43
A.2	Définition de corrélation des noms .....	43
A.3	Définitions de paramètre .....	44
A.4	Définition d'un ensemble.....	45
A.5	Définitions d'attribut.....	45
A.6	Définition de groupe d'attributs.....	45
A.7	Définitions d'action .....	46
A.8	Définitions de notification.....	46
A.9	Définitions relatives au comportement.....	47
A.10	Modules ASN.1 .....	47
	Sommaire.....	49

## NOTE D'INFORMATION

Le tableau suivant indique une liste de Recommandations de la série X.700 élaborées en collaboration avec l'ISO/CEI et qui sont identiques à la Norme internationale correspondante. Ce tableau mentionne les références aux numéros des Normes internationales ISO/CEI ainsi que le titre abrégé de la Recommandation | Norme internationale.

Recommandation du CCITT Norme Internationale ISO/CEI	Titre abrégé
X.700   7498-4 (remarque)	Management Framework
X.701   10040	Aperçu général de la gestion des systèmes
X.710   9595 (remarque)	Définition du service commun de transfert d'informations de gestion
X.711   9596-1 (remarque)	Spécification du protocole commun de transfert d'informations de gestion
X.712   9596-2	CMIP PICS
X.720   10165-1	Modèle d'information de gestion
X.721   10165-2	Définition des informations de gestion
X.722   10165-4	Directives pour la définition des objets gérés
X.730   10164-1	Fonction de gestion des objets
X.731   10164-2	Fonction de gestion d'états
X.732   10164-3	Attributs pour représenter les relations
X.733   10164-4	Fonction de signalisation des alarmes
X.734   10164-5	Event Management Function
X.735   10164-6	Log Control Function
X.736   10164-7	Fonction de signalisation des alarmes de sécurité
X.740   10164-8	Security Audit Trail Function
<p>REMARQUE – Cette Recommandation et la Norme internationale ne sont pas identiques, par contre elles sont alignées au point de vue technique.</p> <p>Les titres abrégés qui figurent en anglais correspondent aux Recommandations du CCITT qui n'ont pas encore été approuvées.</p>	



## NORME INTERNATIONALE

## RECOMMANDATION DU CCITT

**TECHNOLOGIES DE L'INFORMATION – INTERCONNEXION  
DE SYSTÈMES OUVERTS – STRUCTURE DES INFORMATIONS DE  
GESTION: DIRECTIVES POUR LA DÉFINITION DES OBJETS GÉRÉS****1 Domaine d'application**

La présente Recommandation | Norme internationale donne des indications aux rédacteurs de Recommandations et de Normes internationales dans lesquelles figurent des définitions d'objets gérés, afin:

- a) d'inciter à assurer la cohérence entre les définitions d'objets gérés;
- b) de faire en sorte que de telles définitions soient rédigées de manière à être compatibles avec les Recommandations | Normes internationales relatives à la gestion OSI;
- c) de réduire, en déterminant les présentations, procédures et définitions de documents utilisés en commun, les cas où plusieurs groupes de travail effectuent des travaux similaires.

A cet effet, la présente Recommandation | Norme internationale spécifie:

- a) les relations entre les Recommandations et Normes internationales de gestion OSI applicables et les définitions de classes d'objets gérés, et comment ces Recommandations et Normes internationales devraient être utilisées dans les définitions de classes d'objets gérés;
- b) les méthodes qu'il convient d'adopter pour définir les classes d'objets gérés et leurs attributs, notifications, actions et comportement, y compris:
  - 1) un résumé des aspects devant être traités dans la définition;
  - 2) les outils de notation qu'il est recommandé d'utiliser dans la définition;
  - 3) les directives de cohérence que la définition peut suivre;
- c) les relations entre définitions de classes d'objets gérés et protocole de gestion, et quelles définitions relatives au protocole sont requises;
- d) la structure recommandée de la documentation relative aux définitions de classes d'objets gérés.

La présente Recommandation | Norme internationale est applicable au développement de toute Recommandation | Norme internationale définissant:

- a) des informations de gestion devant être transférées ou manipulées au moyen d'un protocole de gestion OSI;
- b) les objets gérés auxquels ces informations se rapportent.

La présente Recommandation | Norme internationale ne précise ni n'implique:

- a) les contraintes de développement des définitions de classes d'objets gérés en termes de fonctionnalité, les Recommandations | Normes internationales auxquelles elles se rapportent, ou les utilisations qui en sont faites dans un environnement de gestion particulier;
- b) les directives pour la définition des ressources; elle ne fournit que les directives concernant la définition des objets gérés qui donnent l'aperçu de gestion des ressources.

## 2 Références normatives

Les Recommandations du CCITT | Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Recommandation | Norme internationale. Au moment de la publication, les éditions indiquées étaient en vigueur. Toute Recommandation | Norme internationale est sujette à révision et les parties prenantes aux accords fondés sur la présente Recommandation | Norme internationale sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations | Normes internationales indiquées ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur. Le Secrétariat du CCITT tient à jour une liste des Recommandations du CCITT actuellement en vigueur.

### 2.1 Recommandations du CCITT | Normes internationales identiques

- Recommandation X.660 du CCITT<sup>1)</sup> | ISO/CEI 9834-1<sup>2)</sup>, *Technologies de l'information – Interconnexion de systèmes ouverts – Procédures pour le fonctionnement des autorités d'enregistrement OSI – Procédures générales.*
- Recommandation X.701 du CCITT (1992) | ISO/CEI 10040: 1992, *Technologies de l'information – Interconnexion de systèmes ouverts – Aperçu général de la gestion des systèmes.*
- Recommandation X.720 du CCITT (1992) | ISO/CEI 10165-1: 1992, *Technologies de l'information – Interconnexion de systèmes ouverts – Structure des informations de gestion – Modèle d'information de gestion.*
- Recommandation X.721 du CCITT (1992) | ISO/CEI 10165-2: 1992, *Technologies de l'information – Interconnexion de systèmes ouverts – Structure des informations de gestion – Définition des informations de gestion.*
- Recommandation X.732 du CCITT (1992) | ISO/CEI 10164-3: 1992, *Technologies de l'information – Interconnexion de systèmes ouverts – Gestion des systèmes – Attributs pour représenter les relations.*
- Recommandation X.733 du CCITT (1992) | ISO/CEI 10164-4: 1992, *Technologies de l'information – Interconnexion de systèmes ouverts – Gestion des systèmes – Fonction rapport d'alarme.*

### 2.2 Paires de Recommandations du CCITT | Normes internationales équivalentes par leur contenu technique

- Recommandation X.200 du CCITT (1988), *Modèle de référence pour l'interconnexion des systèmes ouverts pour les applications du CCITT.*  
ISO 7498: 1984, *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base.*
- Recommandation X.208 du CCITT (1988), *Spécification de la syntaxe abstraite numéro un (ASN.1).*  
ISO/CEI 8824: 1990, *Technologies de l'information – Interconnexion de systèmes ouverts – Spécification de la notation de syntaxe abstraite numéro un (ASN.1).*
- Recommandation X.501 du CCITT (1988), *L'annuaire – Modèles.*  
ISO/CEI 9594-2: 1990, *Technologies de l'information – Interconnexion de systèmes ouverts – L'annuaire – Partie 2: Modèles.*
- Recommandation X.650 du CCITT (1992), *Interconnexion de systèmes ouverts – Modèle de référence de base pour la dénomination et l'adressage.*  
ISO 7498-3: 1989, *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base – Partie 3: Dénomination et adressage.*

<sup>1)</sup> Actuellement à l'état de projet de Recommandation.

<sup>2)</sup> A paraître ultérieurement.

- Recommandation X.700 <sup>1)</sup> du CCITT, *Définition du cadre général de gestion pour l'interconnexion de systèmes ouverts (OSI) pour les applications du CCITT.*  
ISO/CEI 7498-4: 1989, *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base – Partie 4: Cadre général de gestion.*
- Recommandation X.710 du CCITT (1991), *Définition du service commun de transfert d'informations de gestion pour les applications du CCITT.*  
ISO/CEI 9595: 1991, *Technologies de l'information – Interconnexion de systèmes ouverts – Définition du service commun d'informations de gestion.*
- Recommandation X.711 du CCITT (1991), *Spécification du protocole commun de transfert d'informations de gestion pour les applications du CCITT.*  
ISO/CEI 9596-1: 1991, *Technologies de l'information – Interconnexion de systèmes ouverts – Protocole commun d'information de gestion – Partie 1: Spécification.*

### 3 Définitions

Pour les besoins de la présente Recommandation | Norme internationale, les définitions suivantes s'appliquent.

#### 3.1 Définitions relatives au modèle de référence de base

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. X.200 du CCITT | ISO 7498:

- a) connexion (N);
- b) entité (N);
- c) couche (N);
- d) point d'accès au service (N);
- e) système ouvert;
- f) gestion des systèmes.

#### 3.2 Définitions relatives à la dénomination et à l'adressage

La présente Recommandation | Norme internationale utilise le terme suivant, défini dans la Rec. X.650 du CCITT | ISO 7498-3:

- a) sélecteur (N).

#### 3.3 Définitions relatives au cadre général de gestion

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans l'ISO/CEI 7498-4:

- a) objet géré;
- b) opération de couche (N).

#### 3.4 Définitions relatives à l'aperçu général de la gestion des systèmes

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. X.701 du CCITT | ISO/CEI 10040:

- a) agent;
- b) définitions génériques;
- c) classe d'objet géré;
- d) information de gestion;

---

<sup>1)</sup> Actuellement à l'état de projet de Recommandation.

## **ISO/CEI 10165-4 : 1992**

- e) gestionnaire;
- f) protocole de gestion de couche (N);
- g) notification;
- h) type de notification;
- i) opération (de gestion des systèmes);
- j) protocole (d'application) de gestion des systèmes.

### **3.5 Définitions relatives au modèle d'information de gestion**

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. X.720 du CCITT | ISO/CEI 10165-1:

- a) action;
- b) classe réelle;
- c) groupe d'attributs;
- d) identificateur d'attribut;
- e) type d'attribut;
- f) ensemble de valeurs d'attribut;
- g) comportement;
- h) caractéristique;
- i) ensemble conditionnel;
- j) confinement;
- k) héritage;
- l) hiérarchie d'héritage;
- m) objet géré de valeur initiale;
- n) instanciation;
- o) ensemble obligatoire;
- p) héritage multiple;
- q) corrélation des noms;
- r) ensemble;
- s) paramètre;
- t) ensemble de valeurs permises;
- u) nom distinctif relatif;
- v) ensemble de valeurs requises;
- w) spécialisation;
- x) sous-classe;
- y) objet subordonné;
- z) hyperclasse;
- aa) objet supérieur.

### **3.6 Définitions relatives au service commun d'information de gestion (CMIS)**

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. X.710 du CCITT | ISO/CEI 9595:

- a) attribut;
- b) services communs d'information de gestion.

### 3.7 Définitions relatives à l'ASN.1

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. X.208 du CCITT | ISO/CEI 8824:

- a) identificateur d'objet;
- b) type séquence;
- c) type séquence-de;
- d) type ensemble;
- e) type ensemble-de;
- f) sous-type;
- g) type;
- h) nom de référence de type;
- i) nom de référence de valeur.

### 3.8 Définitions complémentaires

**3.8.1 définition de classe d'objets gérés:** ensemble de définitions d'attributs, d'opérations, de notifications et de comportement auquel on a assigné un nom de classe d'objets gérés, justifié par l'emploi d'un modèle de classe d'objets gérés et d'un (ou plusieurs) autre(s) modèle(s), des types définis dans la présente Recommandation | Norme internationale, auxquels le modèle de classe d'objets gérés fait référence directement ou indirectement. La définition d'une classe d'objets gérés comporte tous les éléments de définition hérités de son (ses) hyperclasse(s) ainsi que tous les éléments de définition constituant la (les) spécialisation(s) de cette (ces) hyperclasse(s).

**3.8.2 modèle:** présentation normalisée pour la documentation des corrélations de noms et des définitions de classe d'objets gérés ainsi que de leurs éléments constitutants tels que définitions d'ensembles, paramètres, attributs, groupes d'attributs, comportement, et les actions ou notifications.

**3.8.3 classe d'objets d'annuaire:** classe d'objets définie dans la Rec. X.501 du CCITT | ISO/CEI 9594-2.

## 4 Abréviations

ASN.1	Notation de syntaxe abstraite numéro un ( <i>abstract syntax notation one</i> )
CMIP	Protocole commun d'information de gestion ( <i>common management information protocol</i> )
CMIS	Service commun d'information de gestion ( <i>common management information service</i> )
DMI	Définition des informations de gestion ( <i>definition of management information</i> )
IVMO	Objet géré de valeur initiale ( <i>initial value managed object</i> )
MOCS	Déclaration de conformité d'objet géré ( <i>managed object conformance statement</i> )
OSI	Interconnexion des systèmes ouverts ( <i>open systems interconnection</i> )
PDU	Unité de données de protocole ( <i>protocol data unit</i> )
RDN	Nom distinctif relatif ( <i>relative distinguished name</i> )
SAP	Point d'accès à des services ( <i>service access point</i> )
SAP (N)	Point d'accès à des services (N) [(N)-service-access-point]
SDU	Unité de données de service ( <i>service data unit</i> )
SMI	Structure des informations de gestion ( <i>structure of management information</i> )

## 5 Conventions

Un caractère d'imprimerie particulier est utilisé dans l'ensemble de la présente Recommandation | Norme internationale lorsque le texte fait appel à la **notation ASN.1** ou aux outils de notation définis à l'article 8.

Aucune convention définie extérieurement n'est utilisée dans la présente Recommandation | Norme internationale.

## 6 Questions générales

### 6.1 Intégrité des relations

Lors de la définition des classes d'objets gérés, il importe de tenir compte des situations dans lesquelles des exigences de cohérence seront applicables à des instances de ces classes; par exemple lorsque le comportement d'un objet géré est régi par des règles qui ne dépendent pas seulement de son état propre mais de celui d'autres objets gérés dans le système. Toutes les contraintes de ce type doivent être exprimées par une définition de comportement associé aux classes d'objets gérés en question.

L'opération de suppression est un cas particulier dans lequel les définitions associées à l'instanciation d'un objet géré doivent explicitement appeler des règles de cohérence; pour cette opération, ces règles de cohérence sont spécifiées dans la (les) corrélation(s) de noms associée(s) à la classe d'objets gérés. L'effet d'une opération de suppression doit être défini de manière à bien faire ressortir les circonstances dans lesquelles cette opération est autorisée et quelles en sont les conséquences. La corrélation des noms doit en particulier spécifier si la suppression d'une instance de la classe est autorisée lorsque des objets gérés sont encore confinés et quelles sont les règles qui sont applicables lorsque d'autres relations (hors confinement) existent entre l'objet géré à supprimer et d'autres objets gérés, comme ceux qui peuvent exister par suite de la présence d'attributs de relation tels que définis dans la Rec. X.732 du CCITT | ISO/CEI 10164-3. Les règles de cohérence qui sont appliquées à la suppression doivent être telles que cette opération ne puisse pas donner naissance à des relations non cohérentes. Etant donné que ces règles de cohérence sont spécifiées dans le cadre d'une corrélation de noms, les règles applicables à la suppression d'un objet géré quelconque sont fixées au moment de l'instanciation de cet objet.

### 6.2 Caractéristiques héritées

Le processus d'héritage se traduit par l'inclusion, dans la définition de la classe des objets gérés, de toutes les caractéristiques d'hyperclasse(s) de cette classe. Cette règle est appliquée de manière récursive jusqu'au sommet de la hiérarchie d'héritage. Une classe d'objets gérés quelconque comprendra donc toutes les caractéristiques faisant partie de la définition du sommet, plus toutes les caractéristiques qui seront ajoutées dans le processus de définition de toutes les éventuelles sous-classes faisant partie de la hiérarchie d'héritage de cette classe d'objets gérés.

### 6.3 Proposition d'options

En général, on évite de proposer des options dans les définitions d'objets gérés, la difficulté de l'interfonctionnement croissant avec le nombre d'options. Comme indiqué dans la Rec. X.720 du CCITT | ISO/CEI 10165-1, la définition de classe d'objets gérés peut comporter des ensembles conditionnels présents dans une instance de la classe d'objets gérés si la condition spécifiée est applicable. Il est prévu que les conditions appliquées à ces ensembles se rapportent à des caractéristiques normalisées de la ressource visée par la classe d'objets gérés, ou à des fonctions de gestion facultatives assurées par le système de gestion.

## 6.4 Enregistrement

Le processus de définition des classes d'objets gérés impose l'assignation d'identificateurs globalement uniques, appelés identificateurs d'objet, à divers aspects de la classe d'objets gérés comme le nom de la classe d'objets gérés, les types d'attributs, etc. On utilise les valeurs de ces identificateurs dans les protocoles de gestion pour identifier formellement les aspects des objets gérés et les attributs, opérations et notifications qui leur sont associés. Il est donc indispensable que les organismes ou organisations de normalisation intéressés déterminent ou mettent au point un mécanisme d'enregistrement approprié capable de préciser les valeurs des identificateurs d'objet avant d'établir une définition de classe d'objets gérés. La structure de l'identificateur d'objet et les valeurs des arcs initiaux sont spécifiées dans la Rec. X.208 du CCITT | ISO/CEI 8824; d'autres informations relatives à la mise en place de mécanismes et d'autorités d'enregistrement figurent dans la Rec. X.660 du CCITT | ISO/CEI 9834-1.

Une fois qu'un élément d'information de gestion a été affecté d'une valeur d'identificateur d'objet, il est nécessaire qu'aucune révision de la définition de cet élément ne modifie la sémantique de l'information. En pratique, cela signifie que les modifications rédactionnelles des informations de gestion enregistrées sont permises mais que les définitions ne doivent pas être modifiées d'une façon qui soit visible dans le protocole.

Toutes les valeurs d'identificateur d'objet enregistrées dans les Recommandations | Normes internationales relatives à la gestion des systèmes sont affectées dans l'arc

**{joint-iso-ccitt ms(9)}**

L'affectation des arcs issus de **{joint-iso-ccitt ms(9)}** est définie dans la présente Recommandation | Norme internationale. Les arcs issus de **{joint-iso-ccitt ms(9)}** sont assignés en fonction de chaque norme de gestion de systèmes, comme indiqué au tableau 1.

**Tableau 1 – Affectation des arcs issus de {joint-iso-ccitt ms(9)}**

Arc	Norme
<b>smo(0)</b>	Aperçu général de la gestion des systèmes: Rec. X.701 du CCITT   ISO/CEI 10040
<b>cmip(1)</b>	Protocole commun de transfert d'information de gestion, Rec. X.711 du CCITT   ISO/CEI 9596-1
<b>function(2)</b>	Fonctions de gestion des systèmes; Rec. X.7NN du CCITT   ISO/CEI 10164-X, où X est le numéro de partie de la norme dans le catalogue ISO/CEI et X.7NN le numéro de la Recommandation correspondante du CCITT
<b>smi(3)</b>	Structure des informations de gestion, Rec. X.72N du CCITT   ISO/CEI 10165-X, où X est le numéro de partie de la norme dans le catalogue ISO/CEI et X.72N le numéro de la Recommandation correspondante du CCITT

L'affectation des arcs inférieurs à ce niveau est définie en 6.4.1 à 6.4.5. D'autres arcs imposés par des normes de gestion de systèmes, existantes ou futures, seront affectés, selon les besoins, par des modifications apportées à la présente Recommandation | Norme internationale.

REMARQUE – Le schéma d'affectation des valeurs d'identificateur d'objet défini dans le présent article et ses paragraphes ne s'applique qu'à l'affectation des valeurs d'identificateur d'objet dans les normes de gestion de systèmes rédigées conjointement par le groupe de travail 4 de l'ISO | CEI/JTC1/SC21 et la commission d'études VII du CCITT. Les autres instances ou organismes de normalisation ayant besoin d'affecter des valeurs d'identificateur d'objet pendant l'élaboration de normes de gestion doivent établir leur propre schéma d'affectation sous la supervision d'une autorité d'enregistrement appropriée. Pour établir un schéma d'affectation correct, on peut utilement se référer à la structure adoptée par l'activité de normalisation de la gestion des systèmes, mais le choix final de ce schéma dépend de l'organisation concernée. Afin d'améliorer la lisibilité humaine des valeurs d'identification d'objet, il est recommandé d'utiliser chaque fois que possible, la forme de nom et de numéro définie dans la Rec. X.208 du CCITT | ISO/CEI 8824 pour la représentation des valeurs d'identificateur d'objet.

**6.4.1 Affectation de l'identificateur d'objet pour l'aperçu général de la gestion des systèmes**

REMARQUE – L'aperçu général de la gestion des systèmes est chargé de l'affectation de ces arcs, qui ne sont indiqués que pour information.

En dessous de **{joint-iso-ccitt ms(9) smo(0)}**, les arcs suivants sont affectés à l'enregistrement des identificateurs de contexte d'application, de syntaxe abstraite et de modules ASN.1, comme indiqué au tableau 2.

**Tableau 2 – Affectation des arcs en dessous de {joint-iso-ccitt ms(9) smo(0)}**

Arc	Objectif
<b>applicationContext(0)</b>	Affectation des identificateurs de contexte d'application
<b>negotiationAbstractSyntax(1)</b>	Affectation des identificateurs de version pour la syntaxe abstraite de négociation
<b>asn1Modules(2)</b>	Affectation des identificateurs de module ASN.1

En dessous de **{joint-iso-ccitt ms(9) smo(0) applicationContext(0)}**, d'autres arcs sont affectés conformément aux spécifications de la Rec. X.701 du CCITT | ISO/CEI 10040, à l'enregistrement d'identificateurs de contextes d'application particuliers, comme indiqué dans le tableau 3.

**Tableau 3 – Affectation des arcs en dessous de {joint-iso-ccitt ms(9) smo(0) applicationContext(0)}**

Arc	Objectif
<b>systems-management(2)</b>	Identificateur du contexte d'application de gestion des systèmes

En dessous de **{joint-iso-ccitt ms(9) smo(0) negotiationAbstractSyntax(1)}**, d'autres arcs sont affectés, conformément aux spécifications de la Rec. X.701 du CCITT | ISO/CEI 10040, à l'enregistrement de versions particulières de la syntaxe abstraite de négociation, comme indiqué dans le tableau 4.

**Tableau 4 – Affectation des arcs en dessous de {joint-iso-ccitt ms(9) smo(0) negotiationAbstractSyntax(1)}**

Arc	Objectif
<b>version1(1)</b>	Identificateur de la version 1 de la syntaxe abstraite de négociation

En dessous de **{joint-iso-ccitt ms(9) smo(0) asn1Modules(2)}**, d'autres arcs sont affectés, conformément aux spécifications de la Rec. X.701 du CCITT | ISO/CEI 10040, à l'enregistrement d'identificateurs particuliers de module ASN.1, comme indiqué dans le tableau 5.

**Tableau 5 – Affectation des arcs en dessous de {joint-iso-ccitt ms(9) smo(0) asn1Modules(2)}**

Arc	Objectif
<b>negotiationDefinitions(0)</b>	Affectation d'identificateurs de version au module ASN.1 qui contient les définitions associées à la syntaxe abstraite de négociation

En dessous de **{joint-iso-ccitt ms(9) smo(0) asn1Modules(2) negotiationDefinitions(0)}**, d'autres arcs sont affectés, conformément aux spécifications de la Rec. X.701 du CCITT | ISO/CEI 10040, à l'enregistrement de versions particulières du module ASN.1, comme indiqué dans le tableau 6.

**Tableau 6 – Affectation des arcs en dessous de  
{joint-iso-ccitt ms(9) smo(0) asn1Modules(2) negotiationDefinitions(0)}**

Arc	Objectif
<b>version1(1)</b>	Identification de la version 1 du module ASN.1 qui contient les définitions associées à la syntaxe abstraite de négociation

#### 6.4.2 Affectation de l'identificateur d'objet pour le CMIP

REMARQUE – Le CMIP est responsable de l'affectation de ces arcs, qui ne sont indiqués que pour information. La version 1 du CMIP est périmée et a été remplacée par la version 2. La version 1 était développée dans l'ISO/CEI 9596, qui n'a pas de correspondance dans les Recommandations du CCITT.

En dessous de {joint-iso-ccitt ms(9) cmip(1)}, les arcs sont affectés à chaque version du CMIP, comme indiqué en 6.4.2.1 et 6.4.2.2.

##### 6.4.2.1 Version 1 du CMIP

En dessous de {joint-iso-ccitt ms(9) cmip(1)}, les arcs sont affectés pour la version 1 du CMIP, comme indiqué dans le tableau 7.

**Tableau 7 – Affectation des arcs en dessous de  
{joint-iso-ccitt ms(9) cmip(1)} pour la version 1 du CMIP**

Arc	Objectif
<b>version1(1)</b>	Affectation des identificateurs d'objets pour la version 1 du CMIP

En dessous de {joint-iso-ccitt ms(9) cmip(1) version1(1)}, les arcs sont affectés aux fins décrites dans l'ISO/CEI 9596, comme indiqué dans le tableau 8.

**Tableau 8 – Affectation des arcs en dessous de  
{joint-iso-ccitt ms(9) cmip(1) version1(1)}**

Arc
<b>aAssociateUserInfo(1)</b>
<b>aAbortUserInfo(2)</b>
<b>protocol(3)</b>
<b>abstractSyntax(4)</b>

## 6.4.2.2 Version 2 du CMIP

En dessous de **{joint-iso-ccitt ms(9) cmip(1)}**, les arcs sont affectés pour la version 2 du CMIP, comme indiqué dans le tableau 9

**Tableau 9 – Affectation des arcs en dessous de  
{joint-iso-ccitt ms(9) cmip(1)} pour la version 2 du CMIP**

Arc	Objectif
<b>modules(0)</b>	Affectation des identificateurs d'objets pour les modules ASN.1 du CMIP
<b>cmip-pci(1)</b>	Affectation des identificateurs d'objets pour les informations de commande du protocole CMIP

En dessous de **{joint-iso-ccitt ms(9) cmip(1) modules(0)}**, les arcs sont affectés aux fins décrites dans la Rec. X.711 du CCITT | ISO/CEI 9596-1, comme indiqué dans le tableau 10.

**Tableau 10 – Affectation des arcs en dessous de  
{joint-iso-ccitt ms(9) cmip(1) modules(0)}**

Arc
<b>aAssociateUserInfo(1)</b>
<b>aAbortUserInfo(2)</b>
<b>protocol(3)</b>

En dessous de **{joint-iso-ccitt ms(9) cmip(1) cmip-pci(1)}**, les arcs sont affectés aux fins décrites dans la Rec. X.711 du CCITT | ISO/CEI 9596-1, comme indiqué dans le tableau 11.

**Tableau 11 – Affectation des arcs en dessous de  
{joint-iso-ccitt ms(9) cmip(1) cmip-pci(1)}**

Arc
<b>reserved1(1)</b>
<b>reserved2(2)</b>
<b>reserved3(3)</b>
<b>abstractSyntax(4)</b>

### 6.4.3 Affectation de l'identificateur d'objet pour les normes de fonction

En dessous de **{joint-iso-ccitt ms(9) function(2)}**, les arcs sont affectés de la manière suivante afin d'identifier chaque Recommandation | Norme internationale de fonction comme indiqué dans le tableau 12.

**Tableau 12 – Affectation des arcs en dessous de {joint-iso-ccitt ms(9) function(2)}**

Arc	Document
<b>partX(X)</b>	Rec. X.7NN du CCITT   ISO/CEI 10164-X  où X est le numéro de partie de la fonction de gestion des systèmes dans le catalogue ISO/CEI et X.7NN le numéro de la Recommandation correspondante du CCITT

En dessous de **{joint-iso-ccitt ms(9) function(2) partX(X)}**, les arcs sont affectés comme indiqué dans le tableau 13.

**Tableau 13 – Affectation des arcs en dessous de {joint-iso-ccitt ms(9) function(2) partX(X)}**

Arc	Objectif
<b>standardSpecificExtension(0)</b>	Extensions spécifiques d'une norme, au schéma d'affectation
<b>functionalUnitPackage(1)</b>	Affectation des identificateurs d'ensemble d'unités fonctionnelles
<b>asn1Module(2)</b>	Affectation des identificateurs de module ASN.1
<b>managedObjectClass(3)</b>	Affectation des identificateurs de classe d'objets gérés
<b>package(4)</b>	Affectation des identificateurs d'ensembles
<b>parameter(5)</b>	Affectation des identificateurs de paramètres
<b>nameBinding(6)</b>	Affectation des identificateurs de corrélation de noms
<b>attribute(7)</b>	Affectation des identificateurs d'attributs
<b>attributeGroup(8)</b>	Affectation des identificateurs de groupes d'attributs
<b>action(9)</b>	Affectation des types d'action
<b>notification(10)</b>	Affectation des types de notification

Chaque Recommandation | Norme internationale de fonction peut spécifier, le cas échéant, l'affectation d'autres arcs en dessous de ce niveau (par exemple l'affectation d'identificateurs d'attributs particuliers).

6.4.4 Affectation de l'identificateur d'objet pour les normes SMI

En dessous de {joint-iso-ccitt ms(9) smi(3)}, les arcs sont affectés pour identifier chaque Recommandation | Norme internationale SMI comme indiqué dans le tableau 14.

Tableau 14 – Affectation des arcs en dessous de {joint-iso-ccitt ms(9) smi(3)}

Arc	Document
<b>partX(X)</b>	Rec. X.72N du CCITT   ISO/CEI 10165-X  où X est le numéro de partie de la norme dans le catalogue ISO/CEI et X.72N le numéro de la Recommandation correspondante du CCITT

En dessous de {joint-iso-ccitt ms(9) smi(3) partX(X)}, les arcs sont affectés comme indiqué dans le tableau 15.

Tableau 15 – Affectation des arcs en dessous de {joint-iso-ccitt ms(9) smi(3) partX(X)}

Arc	Objectif
<b>standardSpecificExtension(0)</b>	Extensions spécifiques d'une norme, au schéma d'affectation
<b>asn1Module(2)</b>	Affectation des identificateurs de module ASN.1
<b>managedObjectClass(3)</b>	Affectation des identificateurs de classe d'objets gérés
<b>package(4)</b>	Affectation des identificateurs d'ensemble
<b>parameter(5)</b>	Affectation des identificateurs de paramètre
<b>nameBinding(6)</b>	Affectation des identificateurs de corrélation de noms
<b>attribute(7)</b>	Affectation des identificateurs d'attribut
<b>attributeGroup(8)</b>	Affectation des identificateurs de groupe d'attributs
<b>action(9)</b>	Affectation des types d'action
<b>notification(10)</b>	Affectation des types de notification

Chaque Recommandation | Norme internationale peut, le cas échéant, spécifier que d'autres arcs soient affectés en dessous de ce niveau (par exemple des identificateurs d'attributs particuliers).

#### 6.4.5 Affectation de l'identificateur d'objet pour une classe réelle

La valeur d'identificateur d'objet

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

est affectée par la présente Recommandation | Norme internationale afin de transmettre la sémantique de *classe réelle* qui est définie dans la Rec. X.720 du CCITT | ISO/CEI 10165-1. Lorsque cette valeur d'identificateur d'objet est utilisée pour spécifier la classe d'objets gérés de base dans une demande de service relative à une opération de CMIS, elle indique que le destinataire de l'opération de gestion des systèmes doit répondre en tant qu'élément de sa classe réelle.

### 6.5 Conformité

La Rec. X.701 du CCITT | ISO/CEI 10040 spécifie les règles de conformité générale applicables aux normes relatives aux informations de gestion.

### 6.6 Complexité des définitions des objets gérés

Le processus de modélisation doit permettre de réduire la complexité des définitions des objets gérés. Dans tous les cas, les opérations de gestion ne doivent pas être plus compliquées que les propriétés correspondantes de l'entité OSI en cause.

### 6.7 Création et suppression d'objets gérés

La création et la suppression d'instances d'objets gérés peuvent s'effectuer selon les manières suivantes:

- on peut créer et supprimer des objets gérés par interactions du protocole de gestion. Les opérations de création et de suppression sont définies à cette fin, avec la sémantique associée;
- on peut créer et supprimer des objets gérés en effectuant l'opération de la ressource à laquelle ils se rapportent, généralement une machine protocole. Les opérations de création et de suppression ne seront pas définies dans ce cas. Il s'agira par exemple de la représentation de connexions à des fins de gestion;
- on peut créer et supprimer des objets gérés par d'autres moyens, pour lesquels les opérations de création et de suppression ne sont pas définies. C'est par exemple le cas d'un objet géré qui est toujours créé automatiquement lorsqu'une partie d'équipement est initialisée et qui ne peut pas être supprimé par gestion.

Le choix entre ces trois méthodes peut différer selon qu'il s'agit de créer ou de supprimer des objets gérés.

Dans certains cas, il peut n'y avoir qu'une seule manière de créer ou de supprimer un objet géré d'une classe particulière; dans d'autres cas, il est possible de créer ou de supprimer les objets gérés d'une certaine classe en combinant plusieurs de ces mécanismes.

#### 6.7.1 Objets gérés de valeur initiale

Lorsque l'on crée un objet géré, il est parfois souhaitable de prévoir la possibilité d'affecter des valeurs par défaut à utiliser au moment de la création. Ces valeurs seront elles-mêmes sujettes à modification par opérations de gestion. Cette possibilité peut être assurée par la spécification d'un objet géré de valeur initiale (IVMO) (*initial value managed object*) dont les attributs seront modifiables par opérations de gestion et qui sera capable d'affecter des valeurs par défaut aux attributs correspondant à la création d'instances d'une autre classe d'objets gérés.

Lorsqu'on utilise un IVMO pour créer un nouvel objet géré, les valeurs des attributs contenus dans cet IVMO serviront de valeurs initiales pour les attributs correspondants du nouvel objet géré. La définition de classe de l'objet géré pourra spécifier la manière de choisir l'IVMO. La spécification de l'IVMO doit définir les circonstances dans lesquelles cet objet fournit ses valeurs initiales, comment il le fait et pour quels attributs.

Lorsque l'on utilise des opérations de gestion pour modifier les attributs d'un IVMO, les attributs des objets gérés déjà créés, utilisant les valeurs de cet IVMO, ne sont pas modifiés. De même, les opérations de gestion effectuées sur les attributs d'objets gérés déjà créés, utilisant les valeurs d'un IVMO, n'ont pas d'incidence sur les attributs de celui-ci.

### **6.7.2 Sources des valeurs initiales d'attribut**

Les valeurs initiales d'attribut utilisées lors de la création d'objets gérés proviennent de diverses sources, indiquées dans la Rec. X.720 du CCITT | ISO/CEI 10165-1. Lorsqu'un attribut représente une valeur spécifique qui doit être compatible avec une source sous-jacente, cette valeur est prise comme valeur initiale obligatoire à utiliser au moment de la création d'objet.

## **7 Principes généraux pour la définition des objets gérés**

Les principes généraux décrits ici visent à donner des directives aux définisseurs d'objets gérés et à rendre compatibles les définitions d'objets gérés; c'est pourquoi ils sont invités à appliquer aussi souvent que possible les directives ici présentées.

### **7.1 Eléments communs**

Les définisseurs d'objets gérés doivent s'efforcer d'identifier et d'utiliser comme base les éléments suivants:

- les classes communes d'objets gérés définies dans les Recommandations du CCITT | Normes internationales;
- les classes communes d'objets gérés et autres caractéristiques définies dans la Rec. X.721 du CCITT | ISO/CEI 10165-2.

Les définitions d'objets gérés doivent aussi s'efforcer de prendre en considération et de reprendre les définitions issues d'autres groupes afin d'augmenter la compatibilité des définitions. On pourra atteindre cet objectif en mettant au point des modèles d'application à gérer qui soient communs à un certain nombre de groupes de définisseurs d'objets gérés.

### **7.2 Que gérer?**

La définition des classes d'objets gérés et de leurs composants doit répondre à des spécifications clairement établies concernant les objectifs particuliers de gestion. Ces spécifications doivent normalement comporter la gestion des aspects homologues du protocole d'opération d'une couche ou d'une sous-couche et des problèmes perçus ne faisant pas l'objet d'un rapport spécifique par le fournisseur du service sous-jacent à travers la limite de ce service (par exemple qualité d'un service sous-jacent n'atteignant pas des niveaux acceptables). Il est important de conserver une justification pour chaque objectif de gestion au cours de l'élaboration des spécifications de gestion. Les rapports entre chaque composant d'une spécification de gestion (par exemple, classes d'objets gérés, attributs, opérations, notifications, etc.) et cette justification doivent faire l'objet de commentaires.

Les faits intéressants pour la gestion doivent être enregistrés à l'aide des objets gérés qui représentent les ressources là où ces faits interviennent, c'est-à-dire que si l'on a défini un objet géré représentant une ressource particulière (par exemple une connexion), les informations relatives à cette ressource doivent être renvoyées par les objet(s) géré(s) correspondant(s) et non autrement.

### **7.3 Structuration**

On peut employer un certain nombre de techniques pour représenter la structure des objets gérés, afin de refléter les groupements de données ou de fonctions. Chacune de ces techniques présente des avantages et des inconvénients; le choix de la (ou des) technique(s) convenant le mieux à une spécification particulière dépend d'un certain nombre de critères, indiqués ci-dessous.

Les techniques de structuration décrites dans la Rec. X.720 du CCITT | ISO/CEI 10165-1 comportent les éléments suivants:

- groupes d'attributs;
- sous-classes (spécialisation);
- héritage multiple;
- objets gérés confinés;
- ensembles.

On peut s'appuyer sur des conditions normalisées pour définir les groupements d'attributs, d'opérations et de notifications présents ou absents, comme le choix d'options particulières dans une norme de base. Ces groupements de fonctions, présents ou absents, constituent un tout. Les fonctions peuvent être regroupées à la suite d'une sélection d'option selon la Recommandation | Norme internationale relative à la couche de ressource (par exemple, fourniture de la classe 4 de couche transport); ce qui se traduit par des exigences ou possibilités de gestion complémentaires. Le groupement de fonction peut aussi faire suite à la mise en œuvre d'une fonction de gestion définie (par exemple la comptabilité). Ces groupements de fonctions sont définis à l'aide des techniques d'ensembles conditionnels assurées par le modèle de la classe d'objets gérés.

La présence statique ou dynamique du groupement est un critère important dans le choix des techniques de structuration. Si la présence du groupement est fixée au moment de la spécification, l'utilisation des groupes d'attributs, des sous-classes, de l'héritage multiple ou des objets gérés contenus peut convenir. Si la présence est fixée au moment de la mise en œuvre, de l'installation ou de l'instanciation, l'utilisation des objets gérés confinés ou des ensembles conditionnels peut être appropriée. Si la présence du groupement peut varier pendant la vie des objets gérés de confinement ou d'encapsulation, l'utilisation d'objets gérés confinés, qui sont créés et supprimés de façon dynamique, peut convenir.

Un autre critère consiste à prendre en compte la présence d'instances multiples du groupement dans l'objet géré. Si tel est le cas, il convient d'utiliser la technique des objets gérés confinés; dans les autres cas, on peut utiliser indifféremment l'une des cinq techniques de structuration.

## 7.4 Objets gérés

### 7.4.1 Instanciation des hyperclasses

On peut définir des classes d'objets gérés qui ne sont jamais instanciées afin de constituer une base commune à partir de laquelle des sous-classes pourront être spécialisées: on peut par exemple définir une classe d'objets gérés pour un circuit virtuel générique dont les sous-circuits permanents et commutés pourront former des sous-classes.

Dans certains cas, surtout lorsque l'on définit des sous-classes en vue de réviser une norme, il peut exister des hyperclasses à partir desquelles on pourra créer des instances.

### 7.4.2 Hyperclasses sans restrictions

Les règles de l'héritage limitent le nombre des façons dont on peut modifier les ensembles de valeurs requises et de valeurs permises des attributs d'une classe d'objets gérés, lors de la définition d'une sous-classe de cette classe. De la même manière, ces règles limitent la possibilité d'ajouter des paramètres aux actions et aux notifications. Ces limitations font en sorte que chaque sous-classe soit compatible avec l'hyperclasse.

C'est pourquoi il est utile de prévoir de telles sortes d'extension au moment de définir une classe d'objets gérés qui est censée devenir une hyperclasse par rapport à d'ultérieures classes d'objets gérés. Bien qu'il ne soit pas possible de prévoir et d'assurer toutes les extensions, les techniques suivantes autorisent une grande variété d'extensions tout en assurant la compatibilité:

- définir la syntaxe (type) de chaque attribut afin d'inclure dans celui-ci toutes les valeurs qui pourraient normalement s'intégrer à sa sémantique, même si certaines de ces valeurs ne sont pas requises ou demandées dans l'immédiat;
- insérer des possibilités d'extension dans chaque définition d'action et de notification;

- définir une «hyperclasse sans restrictions» comportant ces éléments sans autres restrictions, en tant que base pour la définition de sous-classes plus restreintes. Dans le cas des attributs, cela implique un ensemble de valeurs requises et un ensemble de valeurs permises égal à la syntaxe d'attribut;
- définir des sous-classes spécifiques de cette hyperclasse sans restrictions, qui imposent les restrictions nécessaires aux attributs, actions et notifications.

Le définisseur des objets gérés peut choisir de ne donner des possibilités d'extension qu'à certains des attributs, actions et notifications d'une hyperclasse sans restrictions.

## 7.5 Attributs

### 7.5.1 Ensembles de valeurs d'attribut

Dans certains cas, des options de la Recommandation | Norme internationale de base permettent à l'ensemble des valeurs d'un attribut de varier en fonction des choix de mise en œuvre. Un exemple caractéristique de cette situation est constitué par le cas où la Recommandation | Norme internationale de base autorise une gamme étendue de tailles de paquets, alors que la mise en œuvre correspondant à cette Recommandation | Norme internationale n'en admet qu'une gamme restreinte. En pareil cas, la définition du comportement de l'attribut doit préciser quelles sont les possibilités.

Il peut être nécessaire de définir des valeurs nulles comme valeurs permises de l'ensemble des valeurs d'un attribut, ou, dans le cas d'attributs d'un IVMO, de définir des valeurs d'attribut auxquelles on fait correspondre une sémantique particulière, par exemple «créer un objet géré dont l'attribut correspondant a une valeur nulle» ou «ignorer le présent attribut en tant qu'origine d'une valeur initiale». Les techniques de définition de ces valeurs comportent la définition de la syntaxe abstraite comme étant un type choix, un choix servant à définir l'ensemble des valeurs normales de l'attribut et d'autres choix à définir les valeurs auxquelles correspond une sémantique particulière.

On peut définir l'ensemble des valeurs autorisées d'un attribut d'un certain nombre de façons:

- en définissant l'ensemble des valeurs de l'attribut de manière statique, en tant que partie de la définition de la classe d'objets gérés;
- en définissant un second attribut, dont la valeur précise l'ensemble des valeurs que l'attribut peut prendre.

La première technique minimise le nombre des définitions d'attribut associées à une classe d'objets gérés; cependant, si l'attribut doit comporter un certain nombre de variantes, la seconde technique permet d'éviter d'avoir à définir des sous-classes multiples pour traiter chaque variante de l'ensemble des valeurs possibles.

### 7.5.2 Types d'attributs

Lorsque l'on utilise les types séquence, séquence de ou ensemble comme types de base dans la définition de la syntaxe de l'attribut, les attributs structurés ne peuvent être utilisés que lorsqu'il n'est pas nécessaire de modifier individuellement des éléments de l'attribut, car ces types ASN.1 correspondent à des types d'attribut à une seule valeur. Lorsqu'il est nécessaire de traiter simultanément plusieurs attributs tout en conservant la possibilité de manipuler chacun d'entre eux individuellement, on peut définir des groupes d'attributs et, au besoin, utiliser des définitions de comportement et d'action pour préciser les relations de dépendance entre les membres du groupe.

REMARQUE – Cela n'implique pas qu'il existe une spécification de comportement propre au groupe d'attributs lui-même, qui ne s'applique pas également aux attributs lorsqu'ils sont traités individuellement.

## 7.6 Relations entre valeurs d'attribut

La valeur d'un attribut peut être limitée selon certaines fonctions d'autres valeurs d'attribut. Toutes les relations de cette nature doivent être identifiées.

Si une valeur d'un attribut est limitée par d'autres attributs dans le *même* objet géré, une synchronisation peut être nécessaire pour une seule opération de gestion, lorsqu'une erreur dans la modification de la valeur d'un ou plusieurs des attributs proches peut entraîner des valeurs illicites pour les attributs proches. Lorsqu'une telle spécification existe, elle doit faire partie de la définition de comportement de la classe d'objets gérés.

Lorsqu'une valeur d'un attribut est limitée par d'autres attributs dans *différents* objets gérés et s'il existe aussi une nécessité de synchronisation, il faut l'inclure dans le comportement associé à la classe d'objets. Dans ce cas, lorsque les objets gérés sont tous dans le même système géré et qu'une seule opération de gestion peut modifier les attributs, cette spécification est renforcée par l'utilisation de la capacité de synchronisation atomique d'objets croisés du CMIS.

Le problème général de la synchronisation de multiples opérations de gestion, de multiples attributs dans différents objets gérés ou de multiples systèmes gérés ne peut pas être résolu au moyen du seul protocole de gestion des systèmes actuels.

## 7.7 Modélisation des SAP

Il est généralement nécessaire de représenter comme une partie de la structure d'objet géré associée aux couches, les relations existant entre les entités de la couche (N), les sélecteurs (N) et les entités de la couche (N+1). Il existe un certain nombre de solutions possibles, par exemple:

- la modélisation des relations sous forme d'informations contenues dans les objets gérés de la couche (N+1);
- la modélisation des relations sous forme d'informations contenues dans les objets gérés de la couche (N);
- la modélisation des relations sous forme d'informations contenues dans les objets gérés n'appartenant à aucune de ces couches, c'est-à-dire dans les objets gérés communs à toutes les couches.

La présente Recommandation | Norme internationale recommande d'adopter la deuxième de ces méthodes, qui consiste à représenter les SAP (N) par des objets gérés individuels, dont les informations d'adressage (et autres) auront la forme d'attributs, ainsi que par des attributs de relation désignant les objets gérés des entités (N) et (N+1) associés aux SAP (N). Pour appuyer les exigences de cohérence imposées aux sélecteurs nécessaires pour que l'adressage OSI ne soit pas ambigu, il est proposé que les objets gérés SAP (N) soient **confinés à l'intérieur** des objets gérés correspondant aux entités (N) auxquelles ils sont liés.

REMARQUE – L'exigence de cohérence dont il est question ici consiste à exiger que l'adresse d'une entité (N) combinée à un sélecteur (N) identifie de façon unique une entité (N+1) [ou un ensemble d'entités (N+1) du même type]. Etant donné que cette exigence repose sur la règle d'unicité d'affectation des valeurs du sélecteur (N) à utiliser dans le contexte d'une entité (N) donnée, on peut satisfaire à cette exigence beaucoup plus simplement si ces informations de sélecteur sont conservées par l'entité (N) et non par les entités (N+1).

## 7.8 Statistiques

### 7.8.1 Cohérence

Les définisseurs d'objets gérés doivent s'efforcer d'assurer la cohérence des statistiques à travers les couches en adoptant certains principes de la Rec. X.200 du CCITT | ISO 7498, notamment le concept d'enregistrement des informations intéressantes pour la gestion au moyen d'objets gérés représentant les ressources auxquelles ces informations se rapportent.

Les statistiques peuvent être enregistrées pour les caractéristiques suivantes de la couche (N):

- erreurs locales;
- échanges réussis entre homologues;
- échecs entre homologues;
- refus de service.

Par exemple, si l'on applique les principes définis précédemment dans la présente Recommandation | Norme internationale à la définition d'une connexion de la Rec. X.200 du CCITT | ISO 7498, on obtient l'identification des statistiques primaires suivantes:

- nombre de connexions d'entité (N) établies avec d'autres entités homologues de la couche (N);
- nombre d'échecs locaux d'établissement de connexion d'entité (N);
- nombre d'échecs de négociation entre homologues pour l'établissement d'une connexion d'entité (N);
- nombre de refus d'établissement de connexion (N-1) du fournisseur du service sous-jacent.

Cet ensemble de statistiques donne une vue cohérente de ce qui se passe dans chaque couche (en mode connexion) sans duplication des compteurs.

REMARQUE – Il est nécessaire d'établir des modèles similaires pour les erreurs, les déconnexions, etc.

### 7.8.2 Compteurs d'unités de données de protocole (PDU)

Les définisseurs d'objets gérés doivent spécifier des compteurs d'unités de données de protocole (PDU) (*protocol data unit*) de couche (N) (et des octets de PDU) plutôt que des SDU (et des octets de SDU).

REMARQUE – Il suffira sans doute de compter le nombre d'octets de PDU dans un nombre limité de couches (N).

### 7.8.3 Chevauchements

Les définisseurs d'objets gérés doivent s'efforcer d'assurer la cohérence et d'éviter une duplication ou un chevauchement inutiles des statistiques, par exemple en conservant le décompte des premières demandes d'envoi de PDU et celui des tentatives répétées d'envoi de PDU, pour ne pas être obligé de faire progresser les deux compteurs au même moment. La somme de ces compteurs donne le nombre total de PDU envoyées.

### 7.8.4 Compteurs non débordants

On recommande d'utiliser des compteurs non débordants, qui permettent d'avoir des observateurs multiples sans être obligé de mettre en œuvre les complexes mécanismes de verrouillage associés à la coordination de la remise à zéro.

### 7.8.5 Compteurs d'événements

Il est nécessaire de conserver le décompte des événements de ressources gérées qui ont pour résultat l'émission d'une notification, car l'envoi d'un message M-EVENT-REPORT par le CMIS peut être supprimé par les discriminateurs de retransmission d'événement.

## 7.9 Compteurs

Pour gérer un compteur, le gestionnaire doit connaître son modulo et déterminer ainsi la valeur pour laquelle le compteur «déborde». Il y a donc quatre possibilités pour définir un compteur:

- le compteur ne déborde jamais;
- le modulo fait partie intégrante de la définition de la classe d'objets gérés;
- le modulo est défini dans un attribut associé;
- le modulo est défini à chaque mise en œuvre et est spécifié dans la déclaration de conformité (MOCS).

REMARQUE – Dans les classes d'objets gérés définies par le JTC1 de l'ISO/CEI, on a adopté, pour les couches 1 à 4, la méthode des compteurs qui ne débordent jamais.

## 7.10 Temporisateurs

Il y a avantage à suivre une spécification commune quant à la précision avec laquelle les systèmes doivent conserver les valeurs d'attribut de temporisation utilisées pour les communications de gestion. Les déclarations de comportement précisent la relation entre ces valeurs d'attribut et le fonctionnement réel des temporisateurs dans le protocole.

REMARQUE – Dans les classes d'objets gérés définies par le JTC1 de l'ISO/CEI, on a utilisé, pour les couches 1 à 4, une représentation en virgule flottante pour exprimer les valeurs de temporisation, avec une longueur de mantisse de 16 bits et une longueur maximale d'exposant de 16 bits, afin de gérer une étendue assez grande sans précision excessive. (Cela n'implique pas qu'une *arithmétique* en virgule flottante doive être mise en œuvre.) Les systèmes sont censés être en mesure de conserver les valeurs avec cette précision. Si les autres contraintes le permettent, on doit accepter une demande visant à régler un attribut de temporisation jusqu'à cette précision.

## 7.11 Mise à jour des attributs

Il y a lieu que les définisseurs de classes d'objets gérés veillent à ce que, lors de la définition d'attributs qui peuvent être mis à jour aussi bien par l'opération de gestion que par le fonctionnement normal de la ressource, les effets d'une mise à jour en concurrence soient précisés. En particulier, l'effet d'une opération de remplacement d'une valeur d'attribut peut être perdu si la ressource met aussi à jour le même attribut.

## 7.12 Précision des attributs

Un système de gestion peut essayer d'imposer à un attribut une valeur de précision plus élevée que le niveau accepté par un système actif. On peut s'approcher de telles valeurs de précision plus élevée, par une approximation de la précision définie.

### 7.13 Identification des objets gérés

Toute définition de classe d'objets gérés pour laquelle des instances peuvent exister doit comporter au moins un attribut pouvant servir d'attribut de dénomination pour l'objet géré. Pour être approprié, cet attribut doit être obligatoire, doit pouvoir être vérifié quant à sa valeur, avoir une sémantique telle que sa valeur puisse rester la même pendant la durée de vie de tout objet géré l'utilisant pour la dénomination et avoir un identificateur dont la valeur désigne de façon unique l'objet géré, à l'exclusion de tous les autres objets désignés par le même objet supérieur.

Lorsqu'un objet géré est supprimé, la valeur attribuée à son attribut de dénomination peut être réutilisée pour identifier des objets gérés ultérieurs créés à l'intérieur du même objet supérieur.

S'il est nécessaire de garantir qu'une instance de classe d'objets gérés reste identifiable parmi toutes les autres instances de cette classe après sa suppression, il faut définir un attribut supplémentaire et l'inclure dans la définition de la classe d'objets. Ce sera un *attribut d'identification unique*, dont la sémantique assure que l'identification unique sera conservée dans le temps. Les classes d'objets gérés pour lesquelles cette exigence n'est pas spécifiée n'ont pas besoin d'inclure un attribut d'identification unique dans leur définition.

L'attribut d'identification unique ne doit être accessible qu'en lecture et, lorsqu'il est inclus dans un objet géré, doit figurer dans les notifications émises par celui-ci.

### 7.14 Notifications

#### 7.14.1 Refus de service

Il ne doit pas être émis de notification concernant le retrait d'un service sous-jacent, car le ou les objets gérés représentant ce service sous-jacent sont chargés de signaler tout motif anormal de cette suppression. Cette procédure est destinée à empêcher un arrêt anormal de remonter dans les couches et d'engendrer des notifications fantômes.

#### 7.14.2 Conservation des informations

Les notifications contiennent des informations associées à un événement qui risquent d'être perdues si l'on ne fait qu'établir des statistiques. Il s'agit par exemple:

- du champ d'en-tête d'une PDU reçue pour laquelle une erreur de protocole a été détectée;
- des statistiques relatives à une connexion qui va bientôt être interrompue, pour une raison spécifique;
- de la temporisation d'une séquence de certains événements.

### 7.15 Utilisation des opérations

La définition d'une classe d'objets gérés doit comprendre les opérations appropriées. Des notifications sont spécifiées pour les invocations émises par le système géré. En ce qui concerne les invocations émises par le système gestionnaire, les opérations sont spécifiées en fonction de leur effet direct sur les objets gérés du système géré, comme suit:

- a) si l'effet direct est la création d'une instance de classe d'objets gérés, on utilise l'opération création (*create*). On n'utilise pas la création pour des activités complexes qui exigent la création coordonnée de nombreux objets gérés, ni lorsque des objets gérés sont créés en corollaire d'une modification d'autres objets gérés, ou si des objets gérés d'appui sont créés à cause d'un changement d'état d'un autre objet géré;
- b) si l'effet direct est la suppression d'un objet géré, on utilise alors l'opération suppression (*delete*);
- c) si l'effet direct est de positionner les valeurs d'un ou plusieurs attributs d'un objet géré sur une ou des valeurs données, on utilise alors l'opération remplacement de la valeur d'attribut (*replace attribute value*);
- d) si l'effet direct est de positionner les valeurs d'un ou plusieurs attributs sur une ou des valeurs par défaut (à condition que ces valeurs par défaut aient été définies), on utilise alors l'opération remplacement par valeur par défaut (*replace with default value*);

- e) si l'effet direct est d'ajouter ou d'enlever certains membres d'un ou de plusieurs attributs d'objet géré valués sur un ensemble, on utilise les opérations adjonction d'un membre (*add member*) ou retrait d'un membre (*remove member*);
- f) si l'effet direct est d'extraire une ou plusieurs valeurs d'attribut d'un objet géré, on utilise l'opération obtention d'une valeur d'attribut (*get attribute value*);
- g) dans les autres cas, par exemple lorsqu'il n'y a pas d'effet direct ou si l'effet direct est une combinaison des effets précités, ou encore s'il y a un autre effet sur l'objet entier, on utilise l'opération action (*action*). Des exemples d'utilisation d'une opération action sont décrits ci-après:
  - 1) lorsqu'il n'est pas possible de définir l'opération à effectuer sur un ensemble d'objets gérés par des techniques de détection et de filtrage en liaison avec les opérations obtention d'une valeur d'attribut, remplacement de la valeur d'attribut, remplacement par valeur par défaut, création, suppression, adjonction de membre ou retrait de membre;
  - 2) lorsqu'il faut créer, au titre d'une action atomique, plusieurs objets gérés;
  - 3) lorsque plusieurs objets sans attributs communs doivent être affectés;
  - 4) lorsque la demande ou la réponse concernant l'opération contient des informations qui ne sont pas modélisées par les attributs des objets gérés.

Les notions d'effet direct et d'effet indirect sont analysées dans la Rec. X.720 du CCITT | ISO/CEI 10165-1.

## 8 Outils de notation pour la définition des objets gérés

### 8.1 Vue d'ensemble des outils de notation

Les modèles (*templates*) définis dans le présent article donnent un ensemble commun d'outils et une notation commune pour représenter différents aspects d'une définition de classe d'objets gérés et de la structure de dénomination associée. Les définitions formelles de chaque modèle sont données en 8.3 à 8.11; les conventions syntaxiques utilisées dans ces définitions formelles sont spécifiées en 8.2. Ces définitions formelles précisent les structures que chaque modèle peut ou doit contenir et l'ordre dans lequel ces structures doivent apparaître dans chaque modèle. L'annexe A donne des exemples d'utilisation de ces outils.

La structure et le comportement d'une classe d'objets gérés sont définis essentiellement au moyen du modèle de classe d'objets gérés. Ce modèle détermine les relations d'héritage existant entre la classe d'objets gérés et d'autres classes d'objets gérés, ainsi que les ensembles de comportement, d'attributs, de notifications et d'opérations qui sont compris dans la définition de la classe d'objets gérés. Pour permettre la réutilisation de certaines parties de la présente Recommandation | Norme internationale dans les spécifications d'autres classes d'objets gérés, des modèles supplémentaires sont définis pour spécifier les attributs (individuels et de groupe), le comportement, les actions, les notifications, les paramètres et les ensembles. Ces autres modèles sont «appelés» par d'autres modèles à l'aide du mécanisme de référencement défini en 9.2; ce mécanisme permet de renvoyer, dans une norme, à des spécifications figurant dans d'autres normes et, par là, d'utiliser des définitions génériques dans les définitions des classes d'objets gérés, en complément des définitions locales. Ces modèles supplémentaires peuvent également être inclus «directement», si on le désire.

La dénomination d'une classe d'objets gérés se définit au moyen du modèle de corrélation de noms. Ce modèle identifie la classe d'objets gérés en cours de dénomination et définit le nom distinctif relatif à utiliser pour nommer les instances de cette classe dans le contexte d'une classe supérieure donnée. Ce modèle sert également pour la spécification des relations existant entre deux classes d'objets à la suite d'une corrélation de noms déterminée.

### 8.2 Conventions utilisées dans les définitions de modèle

Le début du modèle se compose d'un `template-label` et d'un `TEMPLATE-NAME`. Les modèles contiennent une ou plusieurs constructions (*constructs*), chacune d'elles étant désignée par un `CONSTRUCT-NAME` et pouvant avoir un `construct-argument`. Le `construct-argument` peut à son tour se décomposer en un certain nombre d'éléments, selon ce qui est prévu pour la définition de la construction en question. Chaque instance d'utilisation d'un modèle déclare un `template-label` unique permettant à d'autres modèles d'y faire référence et, si la construction `REGISTERED AS` est présente, elle attribue la valeur d'un identificateur d'objet ASN.1 sous laquelle l'instance a été enregistrée. Le caractère point-virgule est utilisé pour indiquer la fin de chaque construction (à l'exception de `REGISTERED AS` et de `DEFINED AS`) et pour marquer la fin d'un modèle.

Afin de simplifier la structure des modèles, par exemple lorsque la même structure syntaxique se répète dans une définition de modèle, on peut élaborer des définitions syntaxiques corrélatives. Lorsque de telles définitions syntaxiques corrélatives sont nécessaires pour compléter la définition du modèle, elles sont introduites par les mots clés

`supporting productions`

à la fin de la définition du modèle et se composent de plusieurs productions ayant la forme:

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

La définition du modèle ou d'autres `supporting productions` peuvent faire référence à la définition à l'aide du `definition-label`; la `syntactic-definition` indique l'extension de la définition à l'aide des conventions syntaxiques définies ci-après dans le présent paragraphe. S'il s'agit d'une `syntactic-definition` qui définit plusieurs chaînes de remplacement, on suppose que les références à la `supporting production` qui la contient ont pour résultat le choix d'une seule chaîne parmi la liste des chaînes possibles.

Les définitions de modèle sont établies à partir des conventions syntaxiques suivantes:

- a) tous les symboles et mots clés de terminaux qui font partie d'un modèle sont sensibles à l'inversion haut/bas de casse;
- b) si cela est nécessaire pour enlever toute ambiguïté à la syntaxe d'un modèle, chaque élément du modèle doit être séparé des éléments adjacents par un ou plusieurs séparateurs. Les séparateurs admis sont le caractère espace, la fin de ligne, un interligne ou un commentaire. On introduit un ou plusieurs séparateurs entre les éléments suivants:

- un `template-label` et un `TEMPLATE-NAME`;
- un `TEMPLATE-NAME` et un `CONSTRUCT-NAME`;
- un `CONSTRUCT-NAME` et un `construct-argument`.

On peut introduire un ou plusieurs séparateurs entre toute autre paire d'éléments dans un modèle, ainsi qu'entre les éléments distincts d'un `construct-argument`. Dans les autres cas, il n'est pas possible d'introduire des séparateurs dans les éléments du modèle, sauf si cela est explicitement admis dans la définition du modèle;

- c) les espaces, interlignes, commentaires et fins de ligne n'ont de valeur qu'en tant que séparateurs;
- d) un commentaire est introduit par le caractère double suivant:

--

et se termine soit par le même caractère double

--

ou par une fin de ligne, si cet élément intervient le premier. Un commentaire est équivalent à un espace lors de l'interprétation des modèles définis dans la présente Recommandation | Norme internationale. Les commentaires n'ont pas de portée normative;

- e) le caractère

;

doit être utilisé pour indiquer la fin de chaque construction dans un modèle (sauf `REGISTERED AS` et `DEFINED AS`) ainsi que la fin d'un modèle;

- f) la notation utilisée pour représenter les identificateurs d'objet doit être la notation de valeur définie dans la Rec. X.208 du CCITT | ISO/CEI 8824 pour la représentation des valeurs d'identificateurs d'objets; c'est-à-dire que la production

`object-identifïer -> <ObjectIdentifierValue>`

est supposée être une production corrélative pour toutes les définitions de modèle données dans la présente Recommandation | Norme internationale, lorsque `ObjectIdentifierValue` renvoie à la notation correspondante définie dans la Rec. X.208 du CCITT | ISO/CEI 8824;

- g) les chaînes entourées par

[ ]

séparent les parties d'une définition de modèle qui peuvent être présentes ou absentes dans chaque instance d'utilisation du modèle. Lorsque le crochet de fermeture est suivi d'un astérisque, c'est-à-dire

[ ]\*

le contenu des crochets peut apparaître zéro fois ou plus. Les circonstances dans lesquelles ces parties de la définition peuvent être omises ou répétées dépendent de la définition du type de modèle;

- h) les chaînes entourées par

< >

sont des chaînes qui doivent être remplacées dans chaque cas d'utilisation d'un modèle. La structure et la signification de la chaîne de remplacement dépendent du type de chaîne;

- i) les chaînes en majuscules indiquent les mots clés qui doivent être présents dans chaque cas d'utilisation d'un modèle, à moins qu'elles ne figurent entre crochets

[ ]

pour indiquer que leur présence est facultative;

- j) le caractère

|

est utilisé comme séparateur entre les chaînes de remplacement dans la `syntactic-definition` d'une `supporting production`. Lorsqu'on utilise une `supporting production` pour définir des chaînes de remplacement, le séparateur d'ouverture de la première chaîne de remplacement est le symbole `->`; le caractère `|` est le séparateur de fermeture et d'ouverture pour les chaînes suivantes et le séparateur de fermeture de la dernière chaîne est la première fin de ligne rencontrée à la suite du séparateur d'ouverture;

- k) un `template-label` doit être unique dans la Recommandation | Norme internationale ou document dans lequel il est déclaré. Dans une Recommandation | Norme internationale ou document dans lequel les différentes parties sont normalement conservées et distribuées séparément, un `template-label` doit être unique dans la partie où il est déclaré.

L'exigence d'unicité des `template-label` ne dépend pas du type du modèle étiqueté. Par exemple, si on utilise dans un document l'étiquette `label1` pour étiqueter une instance d'utilisation d'un modèle, il n'est pas permis d'utiliser l'étiquette `label1` dans le même document pour étiqueter une autre instance d'utilisation d'un modèle du même type ou d'un type différent.

Lorsqu'un `template-label` est déclaré dans le document A et fait l'objet d'une référence dans le document B, cette référence doit être précédée du nom globalement unique du document A. Lorsque le nom du document a été fixé par une autorité de dénomination internationalement reconnue comme le CCITT ou l'ISO/CEI, le désignateur alphanumérique enregistré du document doit servir d'identificateur, par exemple `Rec. X.722 du CCITT (1992) | ISO/CEI 10165-4: 1992`. Le format de cette chaîne doit être conforme à la spécification de l'autorité de dénomination en cause quant aux références à ses documents. Lorsque le document auquel il est fait référence est élaboré et publié conjointement par le CCITT et l'ISO/CEI, le désignateur de ce document doit comporter les deux désignateurs (CCITT et ISO/CEI) du document, séparés par un «|», comme indiqué dans cet exemple. Lorsqu'un nom globalement unique n'est pas encore disponible, il est admissible d'attribuer la valeur d'un identificateur d'objet au document cité en référence et d'utiliser cette valeur comme nom de document globalement unique. La syntaxe d'un `template-label` défini à l'aide de la notation ci-dessus est la suivante:

```
[<document-identif ier> :] <label-string>
```

```
document-identif ier -> "<standard-name>" | <object-identif ier>
```

Un `label-string` peut comporter un nombre quelconque des caractères suivants:

- 1) caractères alphabétiques en majuscules ou en minuscules;
- 2) les chiffres 0 à 9;

3) le caractère (tiret)

-

4) le caractère (barre oblique)

/

dans un ordre quelconque, commençant par un caractère alphabétique minuscule, sauf que le double caractère (deux tirets)

--

ne doit pas figurer dans une chaîne-étiquette. Par exemple, le `template-label` suivant:

```
«Rec. X.722 du CCITT (1992) | ISO/CEI 10165-4: 1992»: exampleObjectClass
```

est une étiquette globalement unique pour la définition de la classe `exampleObjectClass` contenue dans l'annexe A.

Les références d'étiquette non préfixées par un `document-identifiant` sont supposées être des références à des étiquettes déclarées dans le document dans lequel se trouve la référence;

l) chaque fois qu'un `template-label` est présent dans un modèle en tant que pointeur vers un autre modèle, il peut être remplacé par le texte entier du modèle référencé lui-même (y compris l'étiquette du modèle). Cela permet à un modèle d'inclure les modèles auxquels il renvoie (les sous-modèles) «directement» tout en laissant à d'autres modèles la possibilité de faire référence aux sous-modèles ainsi définis. En effet, la *supporting production*

```
template-label -> template-definition
```

est supposée exister pour toutes les instances de `template-label` et de `template-definition`.

m) chaque fois qu'il est nécessaire de faire référence, par un modèle, à une définition de type ou de valeur en ASN.1, le nom de ce type ou de cette valeur ASN.1 est préfixé par le nom du module ASN.1 qui contient la définition de ce type ou de cette valeur. Le nom de module utilisé est supposé se rapporter à un module ASN.1 contenu dans le document contenant lui-même le modèle à partir duquel le type ou la valeur est cité en référence. Les productions corrélatives (*supporting productions*) suivantes:

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

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

sont donc supposées applicables à toutes les définitions de modèle qui font référence à des types ou à des valeurs ASN.1 où `module-name` est le nom attribué à un module ASN.1 contenu dans le document qui fait la référence; et où `type-name` et `value-name` sont les noms attribués aux définitions de type ou de valeur ASN.1 contenues dans ce module. Lorsqu'il faut faire référence à des définitions de type ou de valeur contenues dans d'autres documents, on peut faire appel à un module ASN.1 local qui utilise le mécanisme ASN.1 `IMPORTS` pour importer les définitions appropriées de type ou de valeur;

n) chaque fois qu'il est nécessaire d'introduire un texte dans un modèle, ce texte prend la forme d'une chaîne de caractères commençant et finissant, au besoin, par un caractère séparateur de texte (`text-delimiter`) choisi parmi les caractères suivants:

```
! " # $ % ^ & * ' ` ~ ? @ \
```

Si un caractère `text-delimiter` est utilisé, le même caractère doit être inséré au début et à la fin de la chaîne; chaque fois qu'un caractère `text-delimiter` apparaît dans le corps de la chaîne de texte, il doit être remplacé par deux instances de ce caractère. Si aucun caractère `text-delimiter` n'est utilisé, la chaîne de texte ne doit contenir aucun caractère de ponctuation qui soit dans la suite logique de la chaîne de texte insérée dans la définition du modèle utilisant cette chaîne de texte.

Les productions corrélatives suivantes:

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

```
<text-string>
```

```
text-delimiter -> ! | " | # | $ | % | ^ | & | * | ' | ` | ~ | ? | @ | \
```

sont donc supposées exister pour tous les modèles qui permettent l'emploi d'une chaîne séparée (`delimited-string`), où la chaîne de caractères (`text-string`) est une séquence quelconque de caractères; et si on a utilisé le `text-delimiter`, toutes les instances de ce caractère séparateur dans la `text-string` ont été remplacées par une paire de caractères `text-delimiter`.

A l'exception des règles relatives à l'emploi de séparateurs, la structure interne d'une `text-string`, en particulier l'utilisation de la structure de commentaire définie dans la présente Recommandation | Norme internationale n'a aucune incidence sur les dispositions de celle-ci. Il faut donc supposer que l'ensemble de la `text-string` a une portée normative, sauf indication contraire figurant expressément dans le document où la notation est utilisée.

### 8.3 Modèle de classe d'objets gérés

#### 8.3.1 Vue d'ensemble

Le modèle de classe d'objets gérés forme la base de la définition formelle d'un objet géré. Les éléments contenus dans ce modèle permettent de placer la classe au nœud approprié de l'arbre d'héritage, de spécifier les différents attributs de la classe et de définir le comportement de celle-ci. Les éléments principaux de la définition sont précisés ci-dessous.

##### 8.3.1.1 Héritage

Chaque classe d'objets gérés définit la ou les hyperclasses desquelles elle est extraite. Les sous-classes héritent des caractéristiques de la ou des hyperclasses; la définition de la sous-classe peut s'ajouter à ces caractéristiques (spécialisation) mais elle ne peut supprimer aucune caractéristique de l'hyperclasse. En définitive, toutes les classes sont des sous-classes du sommet de la hiérarchie d'héritage.

##### 8.3.1.2 Ensembles obligatoires

La définition de la classe d'objets gérés comprend les ensembles de comportement, d'attributs, d'opérations et de notifications donnant une spécification complète du comportement commun à toutes les instances de la classe.

##### 8.3.1.3 Ensembles conditionnels

La définition de la classe d'objets gérés comporte la spécification des ensembles de comportement, d'attributs, d'opérations et de notifications présentes ou absentes dans les instances de cette classe à la suite d'une condition spécifiée.

##### 8.3.1.4 Dénomination de classe

La définition de la classe d'objets gérés doit comporter un nom de classe qui peut être utilisé pour se référer à la classe dans le protocole de gestion. Cette procédure s'effectue par enregistrement d'une valeur d'identificateur d'objets identifiant la définition de la classe d'objets gérés.

#### 8.3.2 Structure du modèle

```
<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
```

#### 8.3.3 Définitions corrélatives

##### 8.3.3.1 DERIVED FROM <class-label> [, <class-label>]\*

La construction `DERIVED FROM` doit être présente dans toutes les définitions de classe d'objets gérés autres que le sommet de la hiérarchie d'héritage. C'est par conséquent le cas pour le sommet d'être une hyperclasse de toutes les classes autres qu'elle-même.

La `class-label` identifie une classe d'objets gérés d'où provient la classe d'objets gérés, c'est-à-dire une classe d'objets gérés qui est une des hyperclasses immédiates de la classe d'objets gérés. Un héritage multiple étant autorisé, une classe d'objets gérés peut avoir plusieurs hyperclasses immédiates.

Le processus d'héritage (spécialisation) exige que toutes les caractéristiques de la (ou des) hyperclasse(s) soient incluses dans la définition de la sous-classe.

Les caractéristiques héritées d'une hyperclasse ne doivent pas être répétées dans la documentation de la sous-classe à moins que ne soit utilisée une des techniques décrites dans la présente Recommandation | Norme internationale pour compléter ou modifier une définition héritée d'une hyperclasse. La construction DERIVED FROM est par conséquent censée comporter automatiquement toutes les caractéristiques de la ou des définitions de l'hyperclasse. Ces caractéristiques peuvent être augmentées ou modifiées par les éléments définis dans les constructions CHARACTERIZED BY et CONDITIONAL PACKAGES.

REMARQUE 1 – Il y a lieu d'inclure dans la documentation de la définition de classe d'objets gérés, sous forme de commentaire, une liste de toutes les classes d'objets gérés dont les caractéristiques seront héritées par la définition de cette classe d'objets gérés.

Lorsqu'un héritage multiple se traduit par l'importation multiple de la définition du même élément (ce qui peut, par exemple, se produire si deux définitions d'hyperclasse comprennent le même attribut), on suppose que la sous-classe ne contient qu'une fois la définition en question.

Pour ce qui est de la solution des litiges éventuels entre les éléments définis dans les ensembles et les ensembles conditionnels hérités ou inclus dans la définition de la classe d'objets par spécialisation, tous les ensembles destinés à être inclus dans une classe d'objets donnée sont traités de façon identique. Chaque ensemble définit plusieurs éléments qui sont traités comme suit:

- a) BEHAVIOUR (*comportement*). Les ensembles inclus dans une sous-classe étendent le comportement hérité. Le comportement d'une classe d'objets gérés doit être exprimé de manière à tenir compte de la présence ou de l'absence possible d'ensembles conditionnels.

REMARQUE 2 – Dans certains cas, on peut définir des sous-classes dans lesquelles aucune définition de comportement n'est nécessaire en dehors de celles qui sont héritées de la ou des hyperclasses.

- b) ATTRIBUTES. Des attributs peuvent être spécifiés dans les ensembles inclus dans la définition de la sous-classe. Quand la construction ATTRIBUTES d'un ensemble identifie un attribut qui fait l'objet de définitions multiples dans la classe d'objets gérés, les règles suivantes s'appliquent:
  - 1) un seul attribut de ce type doit être inclus dans l'objet géré instancié;
  - 2) la *propertylist* (*liste de propriétés*) qui en résulte est le OU logique de la *propertylist* incluse dans la sous-classe et la ou les *propertylist* héritées, à l'exception de PERMITTED VALUES (*valeurs permises*) où l'ensemble des valeurs permises instanciées est l'intersection des ensembles de toutes les spécifications de valeurs permises pour ce type d'attribut et de REQUIRED VALUES (*valeurs requises*) où l'ensemble des valeurs requises instanciées est l'intersection des ensembles de valeurs permises instanciées avec la réunion des ensembles de toutes les spécifications de valeurs requises pour ce type d'attribut. Pour les ensembles DEFAULT VALUE ou INITIAL VALUE, si des valeurs litigieuses sont spécifiées pour l'attribut dans les définitions réunies, l'insertion d'un ensemble dans la sous-classe doit résoudre le litige;
  - 3) les paramètres associés à un attribut donné sont représentés par la réunion de l'ensemble de tous les paramètres associés au modèle d'attribut et de l'ensemble de tous les paramètres associés à l'attribut lors de chaque instanciation d'un ensemble.

Si la classe d'objets gérés est destinée à être instanciée, un attribut au moins sera défini dans la définition de la classe, du fait qu'il est nécessaire d'identifier un attribut qui peut servir à nommer des instances d'un objet géré.

REMARQUE 3 – Les attributs servant à nommer peuvent être choisis à partir d'attributs qui font partie de la définition de classe. Cela inclut tous les attributs qui ont été hérités d'hyperclasses et tous ceux qui ont été ajoutés à la classe par le processus de spécialisation.

- c) ATTRIBUTE GROUPS (*groupes d'attributs*). Pour un groupe d'attributs extensible, l'appartenance d'un groupe d'attributs à une instance de la sous-classe est la réunion de l'ensemble de tous les attributs définis dans le modèle de groupe d'attributs et de l'ensemble de tous les attributs ajoutés à ce groupe dans la ou les hyperclasses ou dans la sous-classe.
- d) ACTIONS. Des actions peuvent être incluses dans la définition de la sous-classe; il peut s'agir d'actions qui s'ajoutent à celles qui sont héritées d'hyperclasses, ou encore d'actions comprenant des paramètres supplémentaires pour une action héritée. Les paramètres associés à une action donnée sont la réunion de l'ensemble de tous les paramètres associés au modèle d'action et de l'ensemble de tous les paramètres associés à l'action dans tout ensemble qui est instancié.

- e) NOTIFICATIONS. Dans la définition de la sous-classe, on peut inclure des notifications qui peuvent s'ajouter à celles qui sont héritées des hyperclasses ou qui peuvent comprendre des paramètres supplémentaires pour une notification héritée. Les paramètres associés à une notification donnée sont la réunion de l'ensemble de tous les paramètres associés au modèle de notification et de l'ensemble de tous les paramètres associés à la notification dans tout ensemble qui est instancié.

Si un ensemble est inséré plusieurs fois dans une définition de classe d'objets gérés par héritage et(ou) par inclusions multiples dans le modèle de classe d'objets gérés, la définition de condition associée à l'ensemble sera le OU logique de toutes les condition-définitions contenues dans l'ensemble des définies réunies. A cette fin, les ensembles inclus dans les constructions CHARACTERIZED BY (ensembles obligatoires) sont traités comme s'ils avaient été inclus dans une construction CONDITIONAL PACKAGES avec une définition de condition du type PRESENT IF ! TRUE !.

Les caractéristiques insérées dans un ensemble (obligatoire ou conditionnel) ne doivent dépendre des caractéristiques d'autres ensembles conditionnels que si les conditions de l'ensemble associé sont telles que les caractéristiques requises seront présentes dans tous les objets gérés lorsque le premier ensemble sera présent.

### 8.3.3.2 CHARACTERIZED BY <package-label> [,<package-label>]\*

Cette construction, si elle est présente, permet d'inclure un ou plusieurs ensembles obligatoires de comportement, d'attributs, d'opérations et de notifications dans la définition de la classe d'objets gérés, en plus de ceux qui font partie de la définition en raison de la construction DERIVED FROM (*dérivé de*). Le package-label (*étiquette d'ensemble*) identifie la définition de l'ensemble qui doit être incluse. Le fait de spécifier l'étiquette d'un ensemble qui figure également dans la définition de classe d'objets gérés en tant qu'ensemble conditionnel rend cet ensemble obligatoire dans cette classe d'objets gérés et dans ses sous-classes.

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

Cette construction est présente si un ou plusieurs ensembles conditionnels doivent être inclus dans la classe. Le package-label identifie la définition de l'ensemble qui s'applique. La condition-definition est une description de la condition qui, si elle est vraie, exige que l'ensemble soit inclus dans une instance de la classe. Aucune restriction n'est imposée à la structure ou au jeu de caractères utilisés pour représenter la condition-definition. La condition doit satisfaire aux conditions requises pour les ensembles conditionnels présentés dans la Rec. X.720 du CCITT | ISO/CEI 10165-1. Par exemple

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

constituerait une déclaration d'ensemble valable pour autant que la norme ISO/CEI XXXX définisse l'opération de classe 4 comme une caractéristique facultative valable de la ressource.

### 8.3.3.4 REGISTERED AS object-identifiant

La valeur de object-identifiant fournit un identificateur globalement unique pour la définition de la classe d'objets. Cette valeur est utilisée dans le protocole de gestion quand il est nécessaire d'identifier la classe d'objets.

## 8.4 Modèle d'ensemble

### 8.4.1 Vue d'ensemble

Ce modèle permet de définir un ensemble composé d'une combinaison de définitions de comportement, d'attributs, de groupes d'attributs, d'opérations et de notifications en vue de leur insertion dans un modèle subséquent de classe d'objets gérés au moyen des constructions CHARACTERIZED BY ou CONDITIONAL PACKAGES. Les principaux éléments de la définition sont décrits ci-dessous.

#### 8.4.1.1 Comportement

La définition de l'ensemble fournit une spécification complète du comportement inclus dans cet ensemble. Cela comprend:

- l'effet des opérations sur un objet géré et les circonstances dans lesquelles des notifications sont émises;
- les contraintes éventuellement imposées aux opérations afin de satisfaire aux règles de cohérence, en particulier celles en vertu desquelles la création et la suppression d'objets gérés peuvent être effectuées avec les conséquences de ces opérations;

- une spécification de la manière dont des instances d'une classe d'objets gérés interagissent avec d'autres objets gérés connexes de la même classe ou d'une ou plusieurs autres classes;
- l'identification de tous attributs correspondant à des informations éventuellement insérées dans les notifications. Cela implique d'identifier toute mise en correspondance possible avec des champs de notification particuliers ou avec l'émission d'une notification;
- la spécification de critères de sélection d'IVMO, le cas échéant;
- une définition complète d'autres aspects éventuels du comportement de la classe d'objets gérés.

#### 8.4.1.2 Attributs confinés

L'ensemble d'attributs que l'ensemble contient doit être spécifié.

#### 8.4.1.3 Opérations et notifications

La définition de l'ensemble doit spécifier les notifications que les instances de la classe qui utilisent cet ensemble doivent être capables d'émettre, les opérations que les instances de la classe doivent être capables d'opérer et dans le cas d'opérations relatives aux attributs, les attributs qui doivent être disponibles pour faire l'objet d'opérations. La définition de l'ensemble doit également spécifier tout paramètre additionnel que les notifications et les opérations que les instances de la classe d'objet géré qui utilisent cet ensemble doivent être capables de transporter.

#### REMARQUES

1 Les opérations identifiées dans la définition de classe sont les types d'opérations définis dans la Rec. X.720 du CCITT | ISO/CEI 10165-1 (obtention d'une valeur d'attribut, remplacement de la valeur d'attribut, remplacement par valeur par défaut, etc.). Dans le cas d'actions et de notifications, d'autres définitions sont nécessaires pour caractériser leur fonctionnement, comme décrit en 8.10 et 8.11. Les opérations de création et de suppression sont spécifiées dans le cadre du modèle de corrélation des noms décrit en 8.6, du fait que la création et la suppression d'un objet géré sont étroitement liées à la relation de confinement entre objets supérieurs et subordonnés, plutôt qu'à toutes les instances d'une classe d'objets gérés.

2 La corrélation différée, c'est-à-dire l'affectation de paramètres additionnels à des actions ou à des notifications d'une classe d'objets gérés peut être obtenue par l'insertion, dans cette classe d'objets gérés d'un ensemble contenant (seulement) les actions ou notifications affectées ainsi que leurs nouveaux paramètres. La règle de réunion d'ensembles indiquée en 8.3.3 pour les paramètres d'actions et de notifications implique que le ou les paramètres additionnels soient associés à ces actions ou notifications lors de l'instanciation de l'ensemble contenant celle-ci.

#### 8.4.2 Structure du modèle

```

<package-label>    PACKAGE
    [BEHAVIOUR      <behaviour-definition-label> [,<behaviour-definition-label>]* ;
    ]
    [ATTRIBUTES    <attribute-label> propertylist [<parameter-label>]*
    [,<attribute-label> propertylist [<parameter-label>]*]* ;
    ]
    [ATTRIBUTE GROUPS <group-label> [<attribute-label>]* [,<group-label>
    [<attribute-label>]*]* ;
    ]
    [ACTIONS       <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-spezifier]
                  [INITIAL VALUE          value-spezifier]
                  [PERMITTED VALUES      type-reference]
                  [REQUIRED VALUES       type-reference]
                  [get-replace]
                  [add-remove]
value-spezifier  -> value-reference | DERIVATION RULE <behaviour-definition-label>
get-replace->   GET | REPLACE | GET-REPLACE
add-remove      -> ADD | REMOVE | ADD-REMOVE

```

### 8.4.3 Définitions corrélatives

#### 8.4.3.1 BEHAVIOUR <behaviour-definition-label> [, <behaviour-definition-label>]\*

La construction BEHAVIOUR (*comportement*) permet de décrire complètement le comportement (sémantique) associé à l'ensemble. Cette construction met en rapport la description externe d'un objet géré (ses opérations et notifications) avec son opération interne. Le *behaviour-definition-label* (*étiquette de définition de comportement*) identifie une instance d'utilisation du modèle de comportement. Dans certains cas, on peut définir des ensembles qui n'exigent aucune définition de comportement.

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

Cette construction permet d'inclure des attributs dans la définition de l'ensemble. La *propertylist* (*liste de propriétés*) qui suit chaque *attribute-label* (*étiquette d'attribut*) définit l'ensemble des opérations qui peuvent être effectuées sur l'objet géré par rapport à l'attribut et définit toute valeur par défaut permise ou exigée associée à cet attribut.

La propriété REPLACE-WITH-DEFAULT (*remplacer par valeur par défaut*) est incluse si l'attribut a une valeur par défaut qui peut être fixée au moyen de l'opération *replace with default value* (*remplacer par valeur par défaut*).

La propriété DEFAULT VALUE (*valeur par défaut*) est incluse si l'attribut a une valeur par défaut qui est utilisée pour indiquer la valeur de l'attribut lorsque l'ensemble est instancié conformément aux règles définies dans la Rec. X.720 du CCITT | ISO/CEI 10165-1. Si aucune valeur par défaut n'est spécifiée et que la propriété REPLACE-WITH-DEFAULT est présente, la valeur par défaut est déterminée par d'autres moyens propres au système géré. Cette valeur peut être spécifiée soit au moyen d'une *value-reference* ou au moyen d'une DERIVATION RULE qui spécifiera la manière de déterminer la valeur par défaut.

La propriété INITIAL VALUE (*valeur initiale*) est incluse si l'attribut a une valeur initiale obligatoire; elle sera utilisée au moment de la création pour donner la valeur initiale de l'attribut. Cette valeur peut être spécifiée soit au moyen d'une *value-reference* ou au moyen d'une DERIVATION RULE qui indique comment la valeur initiale doit être déterminée.

Si la propriété PERMITTED VALUES est présente, la *type-reference* (*référence du type*) spécifiera les restrictions éventuellement imposées aux valeurs que l'attribut peut prendre. La forme de la spécification visée par la référence doit être un sous-type du type de syntaxe d'attribut défini au moyen de la notation ASN.1 de sous-type.

REMARQUE 1 – La construction PERMITTED VALUES (*valeurs permises*) n'est nécessaire que dans les définitions d'attributs pour spécifier une restriction sur l'ensemble des valeurs permises par la spécification de la syntaxe d'attribut, par exemple en cas de modification d'une spécification d'attributs existante. Ces restrictions sur l'ensemble des valeurs d'un attribut ne doivent être faites que dans les cas où la restriction est fondée sur une limitation inhérente à la sémantique de l'attribut et non sur une hypothèse arbitraire au sujet de ce qui pourrait constituer un ensemble de valeurs raisonnables.

Si la propriété REQUIRED VALUES (*valeurs requises*) est présente, la *type-reference* spécifie les valeurs éventuelles que l'attribut devra être capable de prendre. La forme de spécification visée par la référence doit être un sous-type du type de syntaxe d'attribut défini au moyen de la notation ASN.1 de sous-type.

REMARQUE 2 – Cette propriété définit l'ensemble des valeurs nécessaires pour la conformité. Par exemple, un objet géré de modem pourrait avoir un attribut de débit avec un ensemble de valeurs admissibles de 0 à 19,2 K; or, une conformité avec la norme de modem pourrait exiger que soit admis un débit particulier parmi l'ensemble des valeurs permises. Comme avec PERMITTED VALUES, ces exigences d'admission d'ensemble de valeurs d'un attribut ne doivent être formulées que dans les cas où l'exigence repose sur une caractéristique inhérente à la sémantique de l'attribut et non sur une hypothèse arbitraire à propos de ce qui pourrait constituer un ensemble de valeurs minimales raisonnables.

Si elles sont présentes, les *parameter-label* (*étiquettes de paramètre*) désignent des paramètres d'erreur particuliers à la classe des objets gérés et associés aux opérations de gestion sur l'attribut. Ces erreurs sont signalées sous la forme de défaillances du traitement. Les modèles appelés en référence définissent la syntaxe de ces paramètres d'erreur.

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

Cette construction permet d'identifier un ensemble de groupes d'attributs comme faisant partie de l'ensemble. La définition originale d'un groupe d'attributs peut être complétée par l'adjonction d'autres *attribute-label*, s'il s'agit d'un groupe d'attributs extensible.

#### 8.4.3.4 ACTIONS <action-label> [<parameter-label>]\* [,<action-label> [<parameter-label>]\*]\*

Si elles sont présentes, les *action-label* (*étiquettes d'action*) identifient l'ensemble des définitions d'action qui sont incluses dans l'ensemble. Les définitions de comportement doivent spécifier l'effet de ces actions sur les objets gérés.

Si elles sont présentes, les *parameter-label* désignent tous paramètres d'information ou de réponse relative à une action, particuliers à la classe des objets gérés, ou tous paramètres d'erreur particuliers à la classe des objets gérés et associés à l'action. Les modèles appelés en référence définissent la syntaxe de ces paramètres.

#### 8.4.3.5 NOTIFICATIONS <notification-label> [<parameter-label>]\* [,<notification-label> [<parameter-label>]\*]\*

Cette construction est présente si des notifications sont incluses dans l'ensemble. Les *notification-label* (*étiquettes de notification*) identifient les définitions de notification qui s'appliquent. Les définitions de comportement doivent spécifier les circonstances dans lesquelles ces notifications sont émises par un objet géré.

Si elles sont présentes, les *parameter-label* désignent tous paramètres d'information ou de réponse relative à une notification, particuliers à la classe des objets gérés, ou tous paramètres d'erreur particuliers à la classe des objets gérés et associés à la notification. Il pourra s'agir de paramètres supplémentaires, utilisés par exemple pour remplir le champ d'information additionnelle des notifications définies dans la Rec. X.733 du CCITT | ISO/CEI 10164-4. Les modèles appelés en référence définissent la syntaxe de ces paramètres.

#### 8.4.3.6 REGISTERED AS object-identifier

La valeur de *object-identifier* (*identificateur d'objet*), si elle est présente, fournit un identificateur globalement unique pour la définition de l'ensemble et enregistre le groupement de définitions de comportement, d'attributs, de groupes d'attributs, d'actions et de notifications que l'ensemble définit. La valeur de *object-identifier* est celle qui est incluse dans l'attribut *ensembles* de toutes les instances de la classe d'objets gérés qui sont créées, conformément aux règles indiquées dans la Rec. X.720 du CCITT | ISO/CEI 10165-1. Cette structure est nécessaire lorsque l'ensemble est appelé en référence par une construction *CONDITIONAL PACKAGES* dans un modèle de classe d'objets gérés.

## 8.5 Modèle de paramètre

### 8.5.1 Vue d'ensemble

Ce modèle permet de spécifier et d'enregistrer des syntaxes de paramètres et le comportement connexe qui peuvent être associés à des attributs particuliers, à des opérations particulières et à des notifications particulières dans le cadre des modèles d'ensemble, d'attribut, d'action et de notification, définis en 8.4, 8.7, 8.10 et 8.11. Le type spécifié dans un modèle de paramètre est utilisé pour compléter une construction *ANY DEFINED BY x* dans une unité de données de protocole (PDU), où *x* est un champ de la PDU qui contient l'identificateur d'objet attribué au paramètre. Ce mécanisme est par exemple applicable à la définition

- des défaillances du traitement;
- des paramètres de demandes/réponses relatives aux actions;
- des paramètres de demandes/réponses relatives aux notifications.

L'utilisation du modèle dans chacun de ces contextes est décrite en 8.5.3.

Les principaux éléments de la définition sont décrits ci-dessous.

#### 8.5.1.1 Définition du contexte

Le modèle spécifie le contexte dans lequel le paramètre est applicable, c'est-à-dire qu'il indique le fait que ce paramètre est inséré dans un champ particulier d'une PDU de gestion.

##### 8.5.1.1.1 Information/réponse relative à une action, information/réponse relative à un événement, erreur spécifique

Lorsque le contexte est identifié sans ambiguïté par la PDU de gestion qui contient le paramètre, ce contexte peut être indiqué par un des cinq mots clés prédéfinis qui sont spécifiés en 8.5.3. Le contexte n'est désigné sans ambiguïté par la PDU de gestion que si, et seulement si, la construction *ANY DEFINED BY* n'apparaît exactement qu'une seule fois dans cette PDU.

8.5.1.1.2 Mot clé de contexte

Si le contexte n'est pas désigné sans ambiguïté par la PDU de gestion qui contient le paramètre, un mot clé de contexte doit être spécifié. Ce mot clé doit indiquer le champ de la PDU de gestion dans laquelle le paramètre peut être contenu.

**Tableau 16 – Utilisation du modèle de paramètre**

Utilisation	Contextes possibles
Construction ATTRIBUTES d'un modèle d'ensemble	SPECIFIC-ERROR
Construction ACTIONS d'un modèle d'ensemble	context-keyword, SPECIFIC-ERROR, ACTION-INFO, ACTION-REPLY
Construction NOTIFICATIONS d'un modèle d'ensemble	context-keyword, SPECIFIC-ERROR, EVENT-INFO, EVENT-REPLY
Construction CREATE d'un modèle de corrélation de noms	SPECIFIC-ERROR
Construction DELETE d'un modèle de corrélation de noms	SPECIFIC-ERROR
Modèle d'attribut	SPECIFIC-ERROR
Modèle d'action	context-keyword, SPECIFIC-ERROR, ACTION-INFO, ACTION-REPLY
Modèle de notification	context-keyword, SPECIFIC-ERROR, EVENT-INFO, EVENT-REPLY

8.5.1.1.3 Utilisation dans d'autres modèles

Le tableau 16 montre comment il est fait référence au modèle de paramètre, avec les options possibles.

Lorsque le paramètre est utilisé comme qualificateur dans une définition d'ensemble, il peut être «en relation différée» avec l'élément qu'il qualifie, c'est-à-dire que d'autres paramètres pourront être ajoutés à une notification déjà définie au moment de la définition de l'ensemble, si la syntaxe de notification est extensible.

### 8.5.1.2 Définition syntaxique

Le modèle de paramètre permet d'associer une syntaxe abstraite à chaque paramètre.

### 8.5.1.3 Référence à un attribut

Au lieu d'indiquer dans le paramètre une définition syntaxique et un enregistrement, le modèle de paramètre peut spécifier ces deux éléments en faisant référence à un modèle d'attribut. L'utilisation de cette construction n'a pas d'incidence sur la signalisation de l'attribut déjà enregistré.

### 8.5.1.4 Comportement

Le modèle définit tout comportement applicable à l'emploi du paramètre qu'il concerne.

## 8.5.2 Structure du modèle

```

<parameter-label> PARAMETER
    CONTEXT                context-type ;
    syntax-or-attribute-choice ;
    [BEHAVIOUR             <behaviour-definition-label>
      [, <behaviour-definition-label>]* ;
    ]
[REGISTERED AS object-identifïer] ;

supporting productions
context-type              -> context-keyword |
                           ACTION-INFO |
                           ACTION-REPLY |
                           EVENT-INFO |
                           EVENT-REPLY |
                           SPECIFIC-ERROR
context-keyword           -> type-reference.<identifïer>
syntax-or-attribute-choice -> WITH SYNTAX  type-reference |
                           ATTRIBUTE    <attribute-label>

```

## 8.5.3 Définitions corrélatives

### 8.5.3.1 CONTEXT context-type

Cette construction définit le contexte dans lequel le paramètre est applicable. Les options sont les suivantes:

- context-keyword: cette option est une référence à un contexte défini à l'extérieur du modèle. La structure de cette référence est composée d'une *type-reference* (*référence de type*) suivie d'un *identifïer* (*identificateur*) qui est le nom d'un champ dans la PDU de gestion spécifiée par la *type-reference*. Cette référence peut donc servir à désigner un contexte défini dans un autre document, par exemple pour indiquer que le paramètre n'est applicable qu'à un champ particulier du paramètre d'information d'événement du CMIS (comme le champ d'information additionnelle dans la Rec. X.733 du CCITT | ISO/CEI 10164-4), ou du paramètre de réponse à l'action du CMIS. Si le paramètre ne correspond pas à nom de champ particulier (par exemple si le paramètre d'information d'événement est défini comme étant un ensemble de paires *identificateur/valeur de paramètre*), un des contextes plus généraux ci-après peut être spécifié;
- ACTION-INFO: cette option définit le paramètre comme étant applicable à la représentation d'un paramètre qui peut être contenu dans le paramètre d'information d'action du CMIS;
- ACTION-REPLY: cette option définit le paramètre comme étant applicable à la représentation d'un paramètre qui peut être contenu dans le paramètre de réponse à une action du CMIS;
- EVENT-INFO: cette option définit le paramètre comme étant applicable à la représentation d'un paramètre qui peut être contenu dans le paramètre de réponse à événement du CMIS;

- EVENT-REPLY: cette option définit le paramètre comme étant applicable à la représentation d'un autre paramètre qui peut lui-même être inséré dans le paramètre de réponse à un événement du CMIS;
- SPECIFIC-ERROR: cette option définit le paramètre comme étant applicable à la représentation ou à l'émission d'une erreur due à une défaillance de traitement dans le CMIS. Lorsque cette option est utilisée avec des paramètres qui s'appliquent à des attributs, la définition de la classe d'objets gérés doit spécifier si d'autres attributs, désignés dans une même demande de remplacement, sont modifiés ou non en cas d'erreur d'opération de remplacement de la valeur d'attribut ou d'opération de remplacement par valeur par défaut d'un certain attribut.

#### 8.5.3.2 WITH SYNTAX type-reference

Cette construction, si elle est présente, indique le type ASN.1 du paramètre, tel qu'il est contenu dans le protocole.

#### 8.5.3.3 ATTRIBUTE <attribute-label>

Cette construction, si elle est présente, indique un modèle d'attribut dont la syntaxe et l'identificateur d'objet servent, respectivement, de syntaxe et d'identificateur pour le paramètre.

#### 8.5.3.4 BEHAVIOUR <behaviour-definition-label> [, <behaviour-definition-label>]\*

Cette construction, si elle est présente, permet de spécifier tout comportement ou toute sémantique associé au paramètre. Si la construction ATTRIBUTE est utilisée, elle ne modifie pas le comportement de l'attribut.

#### 8.5.3.5 REGISTERED AS object-identifier

La valeur de object-identifier (*identificateur d'objet*), si elle est présente, constitue un identificateur globalement unique pour la définition du paramètre. Cette valeur est utilisée dans le protocole de gestion quand il est nécessaire d'identifier le paramètre. Cette structure doit être présente si, et seulement si, la construction WITH SYNTAX est présente.

## 8.6 Modèle de corrélation des noms

### 8.6.1 Vue d'ensemble

Ce modèle permet de définir d'autres structures de dénomination pour les objets gérés d'une certaine classe d'objets gérés, au moyen de corrélations de noms. Une corrélation de noms permet de choisir un attribut comme attribut de dénomination qui devra être utilisé quand un objet subordonné (qui est une instance d'une classe d'objets gérés spécifiée) sera nommé par un objet supérieur qui est une instance d'une classe d'objets gérés spécifiée, ou d'une autre classe d'objets, comme une classe d'objets d'annuaire.

Si une certaine corrélation de noms est utilisée, l'attribut désigné comme attribut de dénomination doit être présent dans l'objet subordonné. L'attribut de dénomination sert à construire le nom distinctif relatif des objets subordonnés de cette classe. Un nom distinctif relatif est construit à partir de l'identificateur d'objet affecté à ce type d'attribut et de la valeur de l'instance de l'attribut. Le nom distinctif de l'objet subordonné est obtenu en ajoutant son nom distinctif relatif au nom distinctif de son objet supérieur.

Les corrélations de noms ne sont pas considérées comme faisant partie de la définition de l'une ou l'autre des classes d'objets gérés qu'elles référencent. Plusieurs corrélations de noms peuvent être associées à une classe d'objets gérés donnée. L'ensemble des corrélations de noms définit l'ensemble des relations de dénomination possibles avec des objets supérieurs et l'ensemble de classes d'objets gérés à partir desquelles des objets subordonnés peuvent être instanciés.

Une corrélation de noms peut également être définie de manière à s'appliquer à toutes les sous-classes de la classe d'objets supérieurs ou de la classe d'objets subordonnés spécifiée, ou des deux classes.

REMARQUE – Des corrélations de noms peuvent être spécifiées pour une classe d'objets gérés après la spécification de cette même classe.

## 8.6.2 Structure du modèle

```

<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> ;
    [BEHAVIOUR                  <behaviour-definition-label>
    [, <behaviour-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

```

## 8.6.3 Définitions corrélatives

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

Cette construction définit une classe d'objets gérés dont les instances peuvent être nommées par des instances de la classe d'objets gérés définie par la construction NAMED BY SUPERIOR OBJECT CLASS (*nommée par une classe d'objets supérieurs*). Le nom d'une instance de cette classe d'objets subordonnés est construit par concaténation du nom distinctif de son objet supérieur avec le nom distinctif relatif de l'objet subordonné. Si la construction AND SUBCLASSES est spécifiée, la corrélation de noms s'applique également à toutes les sous-classes de la classe d'objets gérés spécifiée.

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

Cette construction définit une classe d'objets gérés (ou une autre classe d'objets comme une classe d'objets d'annuaire) dont les instances peuvent appeler des instances de la classe d'objets gérés définie par la construction SUBORDINATE OBJECT CLASS. Si la construction AND SUBCLASSES est spécifiée, la corrélation de noms s'applique également à toutes les sous-classes de la classe d'objets gérés spécifiée.

### 8.6.3.3 WITH ATTRIBUTE <attribute-label>

Cette construction définit l'attribut qui doit être utilisé, dans le contexte de cette corrélation de noms, pour construire le nom distinctif relatif pour les instances de la classe d'objets gérés définie par la construction SUBORDINATE OBJECT CLASS. Les valeurs de cet attribut doivent être représentées par des types de données de valeur unique conformes aux restrictions spécifiées dans la Rec. X.720 du CCITT | ISO/CEI 10165-1; si aucun attribut approprié n'est disponible pour être utilisé comme attribut de dénomination, les concepteurs d'objets gérés sont encouragés à fournir un attribut de dénomination du type `GraphicString` (*chaîne graphique*).

### 8.6.3.4 BEHAVIOUR <behaviour-definition-label> [, <behaviour-definition-label>]\*

Si elle est présente, cette construction permet de définir tout impact sur le comportement imposé par suite de la corrélation des noms. Le `behaviour-definition-label` (*étiquette de définition du comportement*) identifie la définition de comportement en question.

REMARQUE 1 – Cette construction est destinée à servir de moyen pour décrire un comportement propre à une corrélation des noms. Tout comportement applicable à toutes les instances possibles d'une classe d'objets gérés devrait être défini au titre du comportement référencé par le ou les modèles d'ensemble qui définissent la classe d'objets gérés.

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

Cette construction est présente s'il est permis de créer de nouvelles instances de la classe d'objets gérés référencée par la construction SUBORDINATE OBJECT CLASS dans le contexte de cette corrélation des noms, au moyen de l'opération de gestion de systèmes. Les valeurs du `create-modifier` (*modificateur de création*) spécifient les options disponibles pour la création. Les valeurs permises du `create-modifier` sont les suivantes:

- WITH-REFERENCE-OBJECT. Si cette valeur est présente, un objet géré de référence peut être spécifié à la création comme source de valeurs par défaut et pour spécifier le choix d'ensembles conditionnels;
- WITH-AUTOMATIC-INSTANCE-NAMING. Si cette valeur est présente, la demande create (*création*) peut omettre de spécifier le nom d'instance du nouvel objet géré.

Les définitions de comportement doivent spécifier la marche à suivre quand il existe un choix de corrélations de noms qui peuvent être appliquées au nouvel objet géré.

Les sources des valeurs initiales d'attribut utilisées lors de la création d'un objet géré, et leurs valeurs de présence associées, sont définies dans la Rec. X.720 du CCITT | ISO/CEI 10165-1.

Si la valeur `parameter-label` est présente, elle désigne les paramètres d'erreur particuliers à la corrélation de noms et associés à l'opération de création. Ces erreurs sont signalées en tant que défaillances de traitement. Les modèles appelés en référence définissent la syntaxe des paramètres d'erreur.

**8.6.3.6 DELETE [`<delete-modifier>`] [`<parameter-label>`]\***

Cette construction est présente s'il est permis de supprimer des instances de la classe d'objets gérés référencée par la construction SUBORDINATE OBJECT CLASS dans le contexte de cette corrélation de noms. Le `delete-modifier` (*modificateur de suppression*), s'il est présent, indique le comportement d'un objet géré de cette classe si l'objet géré est supprimé. Les `delete-modifier` autorisés sont les suivants:

- ONLY-IF-NO-CONTAINED-OBJECTS. Si cette valeur est présente, les objets gérés confinés doivent être explicitement supprimés par des opérations de gestion avant la suppression de l'objet géré contenant, c'est-à-dire qu'une demande de suppression causera une erreur s'il y a des objets gérés confinés;
- DELETES-CONTAINED-OBJECTS. Si une demande de suppression est appliquée à un objet géré pour lequel le modificateur DELETES-CONTAINED-OBJECTS est spécifié, la demande de suppression n'aboutira pas si un objet géré quelconque, confiné directement ou indirectement, a été soumis au modificateur ONLY-IF-NO-CONTAINED-OBJECTS et possède également un objet géré confiné; sinon, la demande de suppression aboutira en entraînant la suppression des objets gérés en confinement.

D'autres règles, décrivant le comportement relatif à la suppression d'objets gérés en confinement, peuvent être spécifiées dans la construction BEHAVIOUR.

REMARQUE 2 – Etant donné que le modificateur DELETES-CONTAINED-OBJECTS permet de supprimer un objet géré même si celui-ci contient d'autres objets gérés, il est conseillé d'utiliser le modificateur ONLY-IF-NO-CONTAINED-OBJECTS s'il y a un doute quelconque quant au modificateur approprié.

S'il existe des contraintes de suppression relatives à d'autres relations ou conditions qui sont typiques de la classe d'objets gérés, ces contraintes doivent être spécifiées comme faisant partie du comportement de la classe d'objets gérés.

Si la valeur `parameter-label` est présente, elle désigne les paramètres d'erreur particuliers à la corrélation de noms et associés à l'opération de suppression. Ces erreurs sont signalées en tant que défaillances de traitement. Les modèles appelés en référence définissent la syntaxe des paramètres d'erreur.

**8.6.3.7 REGISTERED AS `object-identifiant`**

La valeur de `object-identifiant` (*identificateur d'objet*) fournit un identificateur globalement unique pour la définition de la corrélation des noms. Cette valeur est utilisée pour identifier la corrélation des noms pour les besoins de la gestion de connaissances.

## 8.7 Modèle d'attribut

### 8.7.1 Vue d'ensemble

Ce modèle sert à définir différents types d'attribut. Ces définitions peuvent de plus être combinées par le modèle de groupe d'attributs si des groupes d'attributs sont nécessaires. Les principaux éléments de la définition sont décrits ci-dessous.

#### 8.7.1.1 Dérivation

La définition d'un type d'attribut peut modifier ou gêner la définition d'un autre type d'attribut.

#### 8.7.1.2 Syntaxe d'attribut

La définition d'un type d'attribut doit inclure la définition de la syntaxe qui doit être utilisée pour fournir les valeurs de l'attribut dans le protocole de gestion. Cette définition est réalisée au moyen d'une référence à une définition de type ASN.1. La définition d'une syntaxe d'attribut indique si la valeur d'attribut est un type d'attribut à valeur unique ou valué sur un ensemble. Si le type de base est SET OF (*ensemble-de*), l'attribut est un type valué sur un ensemble; sinon c'est un type à valeur unique.

#### 8.7.1.3 Mise en correspondance des valeurs

La définition d'un type d'attribut peut inclure les modalités valables pour tester la valeur d'une instance du type, c'est-à-dire la question de savoir si l'attribut peut faire l'objet de tests d'égalité, de grandeur, etc. La mise en correspondance des valeurs peut exiger, pour certains types d'attribut, que soit spécifiée la définition du mode de fonctionnement d'une règle de mise en correspondance, dans le cadre de la définition du comportement de l'attribut. L'absence de règles de mise en correspondance dans la définition de l'attribut implique que la mise en correspondance des valeurs n'est pas définie.

#### 8.7.1.4 Comportement

La définition de l'attribut peut inclure la définition d'un comportement propre à cet attribut, c'est-à-dire un comportement qui s'applique à un type d'attribut quelle que soit la classe d'objets gérés qui contient les instances de ce type d'attribut.

#### 8.7.1.5 Identificateur d'attribut

Une valeur d'identificateur d'objet doit être attribuée à chaque attribut à inclure dans la définition d'une classe d'objets gérés. Cette valeur est utilisée dans le protocole de gestion pour identifier l'attribut.

#### 8.7.1.6 Paramètres

La définition d'un attribut peut désigner des paramètres d'erreur particuliers à cet attribut et associés aux opérations de gestion à effectuer sur le type d'attribut. Les modèles appelés en référence mettront ces paramètres en correspondance avec les défaillances de traitement.

### 8.7.2 Structure du modèle

```

<attribute-label> ATTRIBUTE
    derived-or-with-syntax-choice ;
    [MATCHES FOR    qualifieur [, qualifieur]* ;
    ]
    [BEHAVIOUR      <behaviour-definition-label> [, <behaviour-definition-label>]* ;
    ]
    [PARAMETERS    <parameter-label> [, <parameter-label>]* ;
    ]

[REGISTERED AS object-identifier] ;

supporting productions

qualifieur          ->    EQUALITY | ORDERING | SUBSTRINGS |
                        SET-COMPARISON | SET-INTERSECTION
derived-or-with-syntax-choice  ->    DERIVED FROM <attribute-label> |
                                      WITH ATTRIBUTE SYNTAX type-reference

```

### 8.7.3 Définitions corrélatives

#### 8.7.3.1 DERIVED FROM <attribute-label>

Si cette construction est présente, la définition d'attribut prend comme point de départ tous les aspects de la définition référencés par `attribute-label`, y compris ceux qui peuvent être dérivés à leur tour d'autres définitions d'attribut. Les règles d'interprétation de l'effet de la présence de l'un quelconque des autres éléments du modèle d'attribut dans ces circonstances sont les suivantes:

- `MATCHES FOR`: l'ensemble de règles de mise en correspondance qui en résulte doit être le OU logique des règles de mise en correspondance spécifiées par cette construction, avec les règles de mise en correspondance éventuellement dérivées;
- `BEHAVIOUR` est censé étendre toutes définitions de comportement dérivées;
- `REGISTERED AS` est supposé remplacer tout enregistrement dérivé d'autres définitions.

Ce mécanisme de dérivation permet:

- de définir un attribut fondé sur une définition d'attribut existante;
- d'ajouter de nouvelles contraintes à une définition d'attribut existante.

#### 8.7.3.2 WITH ATTRIBUTE SYNTAX `type-reference`

Cette construction, qui n'est présente que si la construction `DERIVED FROM` est absente, identifie le type de données ASN.1 qui décrit comment les instances de la valeur d'attribut sont contenues dans le protocole.

Le type de données ASN.1 définit aussi le type de données de l'attribut lui-même. Si le type de base de la syntaxe est du type ensemble-de, l'attribut est valué sur un ensemble. Tous les autres types de données ASN.1, y compris le type ensemble, le type séquence et le type séquence de, définissent des types d'attribut à valeur unique.

#### 8.7.3.3 MATCHES FOR `qualifier [,qualifier]*`

Cette construction définit les types de test qui peuvent être appliqués à une valeur de l'attribut dans le cadre d'une opération filtre. La mise en correspondance pour la présence d'un attribut est implicitement permise pour tous les attributs. Pour d'autres types de mise en correspondance, si cette construction n'est pas présente, la mise en correspondance n'est pas définie et n'est donc pas permise avec l'attribut. Les options sont les suivantes:

- `EQUALITY` (*égalité*). Dans ce cas, la valeur d'attribut peut faire l'objet de tests d'égalité par rapport à une valeur donnée;
- `ORDERING` (*ordre*). Dans ce cas, la valeur d'attribut peut faire l'objet de tests en fonction d'une valeur donnée, afin de déterminer quelle est celle des deux qui est la plus grande;
- `SUBSTRINGS` (*sous-chaînes*). Dans ce cas, la valeur d'attribut peut faire l'objet de tests en fonction d'une valeur de sous-chaîne donnée afin de déterminer sa présence ou son absence dans la valeur d'attribut;
- `SET-COMPARISON` (*comparaison d'ensembles*). Dans ce cas, la valeur d'attribut peut faire l'objet de tests en fonction d'une valeur donnée afin de déterminer l'égalité d'ensemble ou les relations sous-ensemble/surensemble entre les valeurs;
- `SET-INTERSECTION` (*intersection d'ensembles*). Dans ce cas, la valeur d'attribut peut faire l'objet de tests en fonction d'une valeur donnée afin de déterminer la présence ou l'absence d'une intersection d'ensembles non nulle entre deux valeurs.

#### 8.7.3.4 BEHAVIOUR <behaviour-definition-label> [, <behaviour-definition-label> ]\*

Tout comportement propre à ce type d'attribut peut être défini au moyen de cette construction de comportement. La définition de comportement devra inclure toute spécification supplémentaire nécessaire pour définir les modalités d'application de l'ensemble choisi de règles de mise en correspondance à la définition d'attribut. Le comportement propre à la classe d'objets gérés est défini dans la construction de comportement du modèle de classe d'objets gérés.

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

Les *parameter-label* (*étiquettes de paramètre*) permettent d'associer les paramètres au comportement d'un type d'attribut, pour la définition de défaillances de traitement. Par exemple, un type d'attribut peut manifester une erreur par «violation de contrainte» dans certaines conditions. Un paramètre donnant des informations sur une telle erreur pourra être spécifié au moyen d'un modèle de paramètre avec la construction `CONTEXT SPECIFIC-ERROR` et pourra être appelé en référence par le modèle d'attribut.

**8.7.3.6 REGISTERED AS object-identifiant**

La valeur de *object-identifiant* (*identificateur d'objet*) fournit un identificateur globalement unique pour la définition d'attribut; cela inclut tous les éléments dénotés soit directement ou indirectement par les constructions `DERIVED FROM`, `WITH ATTRIBUTE SYNTAX`, `MATCHES FOR` et `BEHAVIOUR`, lorsqu'elles sont présentes. Cette valeur est utilisée dans le protocole de gestion lorsqu'il est nécessaire d'identifier le type d'attribut. Si cette construction est omise, la définition d'attribut ne doit pas être référencée dans une définition de classe d'objets gérés. Si une définition d'attribut est dérivée à partir d'une définition d'attribut existante qui comporte une construction `REGISTERED AS`, la valeur d'identificateur d'objet attribuée à la définition existante n'est pas un identificateur valable pour la définition obtenue. La construction `REGISTERED AS` doit donc être incluse dans la définition dérivée si celle-ci doit être dénotée en référence dans une définition de classe d'objets gérés.

**8.8 Modèle de groupe d'attributs****8.8.1 Vue d'ensemble**

Ce modèle permet la définition de groupes d'attributs; ces groupes sont applicables aux situations dans lesquelles il est souhaitable d'opérer sur l'ensemble des attributs qui sont membres du groupe. Les définitions de comportement pour une classe d'objets gérés donnée définissent la signification des opérations obtention d'une valeur d'attribut et remplacement par valeur par défaut, appliquées aux groupes d'attributs. Chaque membre du groupe doit être lui-même défini comme un type d'attribut de valeur unique ou valué sur un ensemble.

Le modèle de groupe d'attributs définit l'ensemble minimal des attributs qui constituent le groupe et la valeur d'identificateur d'objet qui est utilisée pour appeler le groupe. Chaque définition de classe d'objets gérés qui référence un groupe d'attributs peut étendre l'appartenance au groupe en ajoutant de nouveaux attributs au groupe; à moins que ce groupe n'ait été défini comme ayant une appartenance fixe; de telles extensions ne s'appliquent qu'aux instances de la classe d'objets dans laquelle l'extension est définie. Les attributs identifiés dans le modèle de groupe d'attributs définissent le nombre minimal de membres du groupe dans toutes les définitions de classe d'objets gérés qui référencent le groupe.

Si un groupe extensible d'attributs est présent dans une définition de classe d'objets, tous les attributs définis pour le groupe, que ce soit dans le corps du modèle de groupe d'attributs ou sous forme d'adjonction au groupe d'attributs par la définition de classe d'objets gérés, doivent être présents dans l'ensemble qui référence le groupe ou dans un des ensembles obligatoires de la classe d'objets gérés.

Si un groupe fixe d'attributs est présent dans une définition de classe d'objets gérés, tous les attributs définis pour le groupe doivent être présents dans l'ensemble qui référence le groupe.

**8.8.2 Structure du modèle**

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

### 8.8.3 Définitions corrélatives

#### 8.8.3.1 GROUP ELEMENTS <attribute-label> [, <attribute-label>]\*

Quand elle est présente, cette construction définit l'ensemble des *attribute-label* (*étiquettes d'attribut*) qui identifient les différents attributs formant les éléments du groupe d'attributs qui doivent être présents dans toutes les instances du groupe d'attributs, chacun d'eux devant être défini au moyen du modèle d'attribut. Les définitions de comportement d'une classe d'objets gérés donnent définissent la signification des opérations d'obtention de valeur d'attribut et de remplacement par valeur par défaut en cas d'application aux groupes d'attributs.

REMARQUE 1 – Cela n'implique pas qu'il existe une spécification de comportement propre au groupe d'attributs lui-même, qui ne s'applique pas aussi aux attributs quand il sont traités individuellement.

Tous les attributs d'un groupe doivent être membres d'une ou de plusieurs définitions de classe d'objets gérés référant le groupe, c'est-à-dire que chaque attribut qui est membre du groupe pour une classe d'objets gérés donnée doit être référencé par la construction ATTRIBUTES dans un ou plusieurs des ensembles référencés par la définition de classe d'objets gérés.

#### 8.8.3.2 FIXED

Quand elle est présente, cette construction indique que le groupe d'attributs a par définition une composition fixe.

#### 8.8.3.3 DESCRIPTION delimited-string

Cette construction permet une description de la sémantique du groupe; par exemple 'Group of all state attributes in the managed object' (*groupe de tous les attributs d'état dans l'objet géré*). Il n'y a pas de restrictions au sujet du jeu des caractères utilisés pour représenter la DESCRIPTION et il n'est pas défini d'autre structure dans la DESCRIPTION.

Cette construction ne doit pas être utilisée comme moyen de définir les aspects relatifs au comportement du groupe ou de ses membres.

#### 8.8.3.4 REGISTERED AS object-identifier

La valeur de *object-identifier* (*identificateur d'objet*) fournit un identificateur globalement unique pour la définition du groupe d'attributs. Cette valeur doit être utilisée dans le protocole de gestion quand il est nécessaire d'identifier le groupe d'attributs. Le groupe d'attributs identifié par cette valeur *object-identifier* dans le contexte d'un objet géré inclut tous les attributs définis dans le corps du modèle de groupe d'attributs, plus, dans le cas d'un groupe extensible d'attributs, ceux qui ont été éventuellement ajoutés au groupe comme conséquence de la définition des éléments du modèle de classe d'objets gérés qui s'applique à l'instanciation de l'objet géré.

REMARQUE 2 – Le modèle de groupe d'attributs définit l'ensemble des attributs (qui peut être l'ensemble vide) qui sont toujours membres du groupe; dans le cas d'un groupe extensible d'attributs, cet ensemble peut être étendu par la construction de groupes d'attributs dans le modèle d'ensemble, afin d'étendre l'ensemble selon les besoins de certaines définitions de classe d'objets gérés. Ce procédé peut convenir quand il est souhaitable de définir un groupe d'attributs dont les membres utilisent en partage une sémantique commune (comme des «attributs d'état»), alors que le nombre d'attributs de ce même type sémantique, qui peuvent être présents dans une classe d'objets gérés donnée, est déterminé au moment de l'instanciation, ou quand différents groupements d'attributs avec la même sémantique sont nécessaires dans des classes d'objets gérés multiples. En général, il n'est possible de déterminer la composition d'un groupe extensible d'attributs qu'au moment de l'instanciation de l'objet géré.

## 8.9 Modèle de comportement

### 8.9.1 Vue d'ensemble

Ce modèle est utilisé pour définir les aspects de comportement des classes d'objets gérés, des corrélations de noms, des paramètres, ainsi que des types d'attribut, d'action et de notification. Le modèle de comportement vise à permettre des dispositions d'extension, mais les spécifications de comportement ne doivent pas changer la sémantique d'une information précédemment définie. Si l'information reste non définie, la définition de comportement doit être explicite au sujet de ce qui n'est pas défini.

#### REMARQUES

1 Il y a lieu d'utiliser les modèles de comportement pour véhiculer des sémantèmes qui ne sont pas entièrement décrits par d'autres modèles. Il convient en particulier que les définisseurs ne se contentent pas des étiquettes pour transmettre des sémantèmes.

2 Une déclaration de comportement est en principe rédigée dans les termes de l'objet géré inclus dans la définition de la classe qui le concerne.

### 8.9.2 Structure du modèle

<behaviour-definition-label> BEHAVIOUR

DEFINED AS delimited-string ;

### 8.9.3 Définitions corrélatives

#### 8.9.3.1 DEFINED AS delimited-string

Le texte de la `delimited-string` (*chaîne séparée*) donne une définition d'un aspect du comportement d'une classe d'objets gérés ou des corrélations de noms, paramètres, attributs, actions ou notifications associés. Cette définition peut être documentée au moyen d'un texte en langage naturel ou de techniques de description formelle. La chaîne peut être une référence textuelle à un article ou à un paragraphe, d'un document ou d'une Recommandation | Norme internationale. Aucune restriction n'est imposée au jeu de caractères utilisé pour représenter la `delimited-string` et aucune autre structure n'est définie dans le texte.

### 8.10 Modèle d'action

#### 8.10.1 Vue d'ensemble

Ce modèle sert à définir le comportement et la syntaxe associés à un type d'action donné. Les types d'action définis au moyen de ce modèle peuvent être contenus dans le service d'action de gestion (M-ACTION) défini dans la Rec. X.710 du CCITT | ISO/CEI 9595. Les principales caractéristiques de la définition sont les suivantes.

##### 8.10.1.1 Comportement

La définition d'un type d'action doit spécifier les fonctions de l'action en ce qui concerne son effet sur la classe d'objets gérés sur laquelle il opère. Quand l'action s'applique à plusieurs classes d'objets gérés, la description du comportement est normalement circonscrite aux caractéristiques de comportement communes à toutes les classes d'objets gérés; un autre comportement spécifique d'une classe d'objets gérés concernant ces actions est décrit dans la définition de la classe d'objets gérés elle-même.

##### 8.10.1.2 Mode de fonctionnement

La définition du type d'action doit indiquer si l'action doit être toujours confirmée, ou si elle peut être confirmée ou non confirmée, au choix du gestionnaire.

##### 8.10.1.3 Syntaxe abstraite

La définition du type d'action doit spécifier toute syntaxe qui doit être utilisée pour transmettre les paramètres d'information d'action et de réponse d'action du service M-ACTION défini dans la Rec. X.710 du CCITT | ISO/CEI 9595. Ces syntaxes sont définies au moyen de types de données ASN.1.

#### REMARQUES

1 Sauf intention expresse d'exclure une future extension des arguments d'une action, il est recommandé de définir de façon extensible les syntaxes d'information et de réponse de l'action en introduisant, à titre de champ facultatif, le paramètre `SET OF ManagementExtension` (tel que défini dans la Rec. X.721 du CCITT | ISO/CEI 10165-2).

2 Il est recommandé de choisir, comme type de données de base pour les syntaxes d'information et de réponse, le type `SEQUENCE`.

##### 8.10.1.4 Identificateurs d'action

La valeur de l'identificateur d'objet associé à la définition du type d'action est utilisée pour identifier le type d'action dans le protocole de gestion.

### 8.10.1.5 Paramètres

La définition du type d'action peut désigner les paramètres d'information relative à l'action ou de réponse relative à l'action, ou des paramètres d'erreur particuliers, associés à ce type d'action.

### 8.10.2 Structure du modèle

```

<action-label> ACTION
    [BEHAVIOUR                <behaviour-definition-label>
                                [, <behaviour-definition-label>]* ;
    ]
    [MODE CONFIRMED ;
    ]
    [PARAMETERS              <parameter-label> [, <parameter-label>]* ;
    ]
    [WITH INFORMATION SYNTAX  type-reference ;
    ]
    [WITH REPLY SYNTAX       type-reference ;
    ]

REGISTERED AS object-identifïer ;
    
```

### 8.10.3 Définitions corrélatives

#### 8.10.3.1 BEHAVIOUR <behaviour-definition-label> [, <behaviour-definition-label>]\*

Quand elle est présente, cette construction définit le comportement de l'action, les paramètres qui doivent être spécifiés avec l'action, les résultats que l'action peut produire et leur signification. Les *behaviour-definition-label* (*étiquettes de définition de comportement*) réfèrent les descriptions de comportement définies par l'utilisation du modèle de comportement.

#### 8.10.3.2 MODE CONFIRMED

Quand cette construction est présente, l'action doit fonctionner dans le mode confirmé. Si elle est absente, l'action peut être confirmée ou non confirmée, au choix du gestionnaire.

#### 8.10.3.3 PARAMETERS <parameter-label> [, <parameter-label>]\*

Les *parameter-label* (*étiquettes de paramètre*) désignent les paramètres d'information ou de réponse au sujet de l'action, ou les défaillances de traitement associées au type d'action. Voir l'exemple en A.7.

#### 8.10.3.4 WITH INFORMATION SYNTAX Mots clés relatifs aux modèles: type-reference

Quand cette construction est présente, la *type-reference* (*référence du type*) désigne le type de données ASN.1 qui décrit la structure du paramètre d'information relative à l'action contenu dans le protocole de gestion. Si elle est absente, aucune information particulière à l'action n'est associée à l'invocation de l'action.

#### 8.10.3.5 WITH REPLY SYNTAX type-reference

Quand cette construction est présente, la *type-reference* (*référence du type*) désigne le type de données ASN.1 qui décrit la structure du paramètre de réponse relative à l'action contenu dans le protocole de gestion. Si elle est absente, aucune information particulière à l'action n'est associée à la réponse relative à l'action.

#### 8.10.3.6 REGISTERED AS object-identifïer

La valeur de *object-identifïer* (*identificateur d'objet*) fournit un identificateur globalement unique pour la définition du type d'action. Cette valeur est utilisée dans le protocole de gestion quand il est nécessaire d'identifier le type d'action.

## 8.11 Modèle de notification

### 8.11.1 Vue d'ensemble

Ce modèle est utilisé pour définir le comportement et la syntaxe associés à un type de notification donné. Les types de notification définis au moyen de ce modèle peuvent être contenus dans les rapports d'événement par le service M-EVENT REPORT (*rapport d'événement de gestion*) défini dans la Rec. X.710 du CCITT | ISO/CEI 9595. La définition a les principales caractéristiques décrites ci-après.

#### 8.11.1.1 Comportement

La définition d'un type de notification doit spécifier les circonstances dans lesquelles une notification du type est émise.

#### 8.11.1.2 Syntaxe abstraite

La définition du type de notification doit spécifier toute syntaxe à utiliser pour transmettre les paramètres d'information et de réponse au sujet de l'événement, contenus dans le service de rapport d'événement de gestion M-EVENT REPORT défini dans la Rec. X.710 du CCITT | ISO/CEI 9595. Le modèle de notification permet aussi d'affecter des valeurs d'attribut à des champs de la syntaxe.

#### REMARQUES

1 A moins qu'il ne soit expressément prévu d'exclure une future extension des arguments d'une notification, il est recommandé de définir de façon extensible les syntaxes d'information et de réponse de la notification en introduisant, à titre de champ facultatif, le paramètre ASN.1 SET OF ManagementExtension (tel que défini dans la Rec. X.721 du CCITT | ISO/CEI 10165-2).

2 Il est recommandé de choisir des données de base du type SEQUENCE pour la syntaxe d'information et de réponse.

#### 8.11.1.3 Dénomination de la notification

La valeur de l'identificateur d'objet associé à la définition de notification est utilisée pour identifier le type d'événement dans le protocole de gestion.

#### 8.11.1.4 Paramètres

La définition du type de notification peut désigner des paramètres d'information ou de réponse au sujet des événements, ou des paramètres d'erreur particuliers et associés à ce type de notification.

### 8.11.2 Structure du modèle

```
<notification-label> NOTIFICATION
    [BEHAVIOUR                <behaviour-definition-label> [, <behaviour-
                                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 ;
```

### 8.11.3 Définitions corrélatives

#### 8.11.3.1 BEHAVIOUR <behaviour-definition-label> [, <behaviour-definition-label>]\*

Cette construction, si elle est présente, définit le comportement de la notification, les données qui doivent être spécifiées avec la notification, les résultats que la notification peut donner et leur signification. La *behaviour-definition-label* (*étiquette de définition du comportement*) référence une description de comportement définie par l'utilisation du modèle de comportement.

#### 8.11.3.2 PARAMETERS <parameter-label> [, <parameter-label>]\*

Les *parameter-label* (*étiquettes de paramètre*) désignent les paramètres d'information ou de réponse au sujet de l'événement, ou des défaillances du traitement associées au type de notification. Voir l'exemple en A.8.

#### 8.11.3.3 WITH INFORMATION SYNTAX type-reference [AND ATTRIBUTE IDS <field-name> <attribute-label> [, <field-name> <attribute-label>]\*]

Si elle est présente, cette construction identifie le type de données ASN.1 qui décrit la structure de l'information de notification contenue dans le protocole de gestion et permet d'associer les identificateurs d'attribut avec les champs nommés de la syntaxe abstraite. Si elle est absente, il n'y a pas d'information spécifique associée à l'invocation de notification. Si l'option AND ATTRIBUTE IDS est présente, le *field-name* (*nom du champ*) doit être une étiquette définie dans la syntaxe abstraite référencée par *type-reference* qui figure dans la construction. Le type de données qui est étiqueté par le *field-name* est utilisé pour transmettre les valeurs de l'attribut référencé par *attribute-label*. Le type de données ASN.1 de l'attribut doit être le même que le type de données référencé par le *field-name*.

On ne peut utiliser comme nom de champ aucune étiquette faisant partie du type SET OF (*ensemble de*) ou SEQUENCE OF (*séquence de*), ni aucune étiquette faisant partie de la définition d'un tel élément, étant donné que les étiquettes insérées dans de telles constructions itératives ne peuvent pas toujours se rapporter sans ambiguïté à une même instance du type de données. De même, on ne peut utiliser comme nom de champ aucune étiquette d'un élément de type CHOICE (*choix*), SET (*ensemble*) ou SEQUENCE (*séquence*), ni aucune étiquette faisant partie de la définition d'un tel élément, si l'élément étiqueté apparaît plusieurs fois dans la définition du type.

#### 8.11.3.4 WITH REPLY SYNTAX type-reference

Si elle est présente, cette construction identifie le type de données ASN.1 qui décrit la structure de la réponse à la notification qui est véhiculée dans le protocole de gestion. Si elle est absente, il n'y aura pas, associée à la réponse à la notification, d'information particulière à cette notification.

La syntaxe de réponse est utilisée lorsque l'on envoie une notification au moyen du mode confirmé du service M-EVENT-REPORT (*rapport d'événement de gestion*) du CMIS. Les confirmations d'événement ne seront pas renvoyées à l'objet géré. Il appartient à l'agent de décider de l'envoi d'une notification dans le mode confirmé ou non confirmé, d'après les politiques associées au gestionnaire. Si la construction WITH REPLY SYNTAX est omise d'une définition de notification mais que celle-ci soit envoyée en mode confirmé, la confirmation ne comportera pas d'information de réponse.

#### 8.11.3.5 REGISTERED AS <object-identifiant>

La valeur de *object-identifiant* (*identificateur d'objet*) fournit un identificateur globalement unique pour la définition du type de notification. Cette valeur est utilisée dans le protocole de gestion quand il est nécessaire d'identifier le type de notification.

## ANNEXE A

## Exemples d'utilisation des directives

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

Les exemples présentés dans cette annexe sont destinés à illustrer l'utilisation de la notation par modèles et n'ont pas pour objet de fournir des définitions utiles pour des mises en œuvre effectives. En particulier, les définitions de comportement liées à ces exemples sont, nécessairement, un peu artificielles. La Rec. X.721 du CCITT | ISO/CEI 10165-2 donne d'autres exemples d'utilisation de la notation qui présentent un intérêt pratique pour la mise au point de définition de classe d'objets gérés.

## A.1 Définition de classe d'objets gérés

exampleObjectClass MANAGED OBJECT CLASS

```

DERIVED FROM          "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
                      1992":top ;
CHARACTERIZED BY      examplePackage2 ;
CONDITIONAL PACKAGES
  examplePackage1     PACKAGE
    ACTIONS            qosResetAction,
                      activate ;
    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! ;

```

```

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

```

REMARQUE – Le présent modèle utilise l'option de documentation en ligne pour décrire l'ensemble conditionnel.

## A.2 Définition de corrélation des noms

exampleNameBinding NAME BINDING

```

SUBORDINATE OBJECT CLASS  exampleObjectClass ;
NAMED BY
SUPERIOR OBJECT CLASS     "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
                          1992":system ;
WITH ATTRIBUTE            objectName ;
BEHAVIOUR
  containmentBehaviour    BEHAVIOUR
    DEFINED AS            !A maximum of 3 instances of exampleObjectClass may be
                          contained in any instance of "CCITT Rec. X.721 | ISO/IEC
                          10165-2":system.!
    ;
  ;
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)} ;

```

REMARQUE – Ce modèle utilise l'option de documentation en ligne pour décrire le comportement.

## A.3 Définitions de paramètre

pDUHeader PARAMETER

```

CONTEXT          EVENT-INFO;
WITH SYNTAX      ParameterModule.PDUString;
BEHAVIOUR
  pDUHeaderBehaviour BEHAVIOUR
    DEFINED AS    !PDU header. Carried in the CMIP eventInfo field.!
  ;
;

```

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

createErrorParameter PARAMETER

```

CONTEXT          SPECIFIC-ERROR ;
WITH SYNTAX      ParameterModule.ErrorInfo1 ;
BEHAVIOUR
  createErrorBehaviour BEHAVIOUR
    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.!
  ;
;

```

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

serviceProviderErrorResponseReason PARAMETER

```

CONTEXT          ACTION-REPLY;
WITH SYNTAX      ParameterModule.ServiceProviderErrorResponseReason;
BEHAVIOUR
  serviceProviderErrorResponseReasonBehaviour BEHAVIOUR
    DEFINED AS    !Returned in the responseParameters field of the CMIP
                  actionReplyInfo if responseCode has the value
                  serviceProviderErrorResponse.!
  ;
;

```

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) parameter(5) sperrorrsp(2)};

REMARQUE – Ces modèles utilisent l'option de documentation en ligne pour décrire le comportement.

**A.4 Définition d'un ensemble**

```

examplePackage2    PACKAGE

                    BEHAVIOUR        exampleClassBehaviour ;
                    ATTRIBUTES       objectName                GET ,
                                      qos-Error-Cause          GET ,
                                      qos-Error-Counter        PERMITTED VALUES
                                                                AttributeModule.QOSCounterRange
                                                                REQUIRED VALUES
                                                                AttributeModule.QOSCounterRange
                                                                GET ;

                    ATTRIBUTE GROUPS  qos-Group ;
                    NOTIFICATIONS     protocolError;

REGISTERED AS {joint-iso-ccitt ms(9) smi(3) part4(4) package(4) examplepack2(1)} ;

```

REMARQUE – Comme ce modèle n'est pas utilisé comme ensemble conditionnel, la construction REGISTERED AS n'est pas absolument nécessaire; mais il est plus facile d'inclure l'enregistrement au moment de la spécification que de l'ajouter ultérieurement s'il devient nécessaire d'utiliser cet ensemble comme ensemble conditionnel.

**A.5 Définitions d'attribut**

```

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 ;
                    BEHAVIOUR             qosErrorBehaviour ;

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 ;
                    BEHAVIOUR             qosCounterBehaviour ;

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

```

**A.6 Définition de groupe d'attributs**

```

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! ;

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

```

## A.7 Définitions d'action

```

qOSResetAction    ACTION

    BEHAVIOUR
        reset BEHAVIOUR
            DEFINED AS    !<Definition of the reset behaviour and its effect on the
                           operation of the managed object, etc..>!
        ;
    ;
    MODE CONFIRMED ;

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

```

REMARQUE – Cette définition d'action utilise l'option de documentation en ligne pour le modèle de comportement d'action. Aucune syntaxe abstraite n'est définie pour l'invocation ou la réponse.

```

activate ACTION

    BEHAVIOUR
        activateBehaviour BEHAVIOUR
            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.!
        ;
    ;
    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 Définitions de notification

```

communicationError NOTIFICATION

    BEHAVIOUR          communicationErrorBehaviour ;
    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

    BEHAVIOUR
        protocolErrorBehaviour BEHAVIOUR
            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.!
        ;
    ;
    PARAMETERS          pDUHeader;
    WITH INFORMATION SYNTAX NotificationModule.ProtocolError;

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

```

REMARQUE – Ce modèle utilise l'option de documentation en ligne pour décrire le comportement.

## A.9 Définitions relatives au comportement

qOSCounterBehaviour BEHAVIOUR

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.! ;

qOSErrorBehaviour BEHAVIOUR

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 behaviour definitions associated with the managed object class definition. !;

communicationErrorBehaviour BEHAVIOUR

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 Behaviour definition could, for example, refer to pieces of specification in a base standard in order to specify the behaviour. !;

exampleClassBehaviour BEHAVIOUR

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

## A.10 Modules ASN.1

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

QOSCounterRange ::= QOSErrorCounter {0..4294967296} -- La gamme est de 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, ManagementExtension

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 ManagementExtension

ProtocolError ::= SET OF ManagementExtension

END

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

DEFINITIONS ::= BEGIN

**IMPORTS**

OperationalState, ManagementExtension

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

ActivateReply ::= SEQUENCE {

operationalStatus	[0] OperationalState,
responseCode	[1] INTEGER {successResponse (0), serviceProviderErrorResponse (1) },
responseParams	[2] SET OF ManagementExtension 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

## Sommaire

	Page
Affectation de l'identificateur d'objet ( <i>Object identifier allocation</i> )	
pour l'aperçu général de la gestion des systèmes .....	8
pour le CMIP .....	9
pour les normes de fonction .....	11
pour les normes SMI .....	12
pour une classe réelle .....	13
Définitions des symboles globaux pour les modèles ( <i>template global symbol definitions</i> )	
delimited-string .....	23
document-identifiant .....	22
object-identifiant .....	21
supporting productions .....	21
text-delimiter .....	23
type-reference .....	23
value-reference .....	23
Étiquettes des modèles ( <i>template labels</i> )	
action-label .....	40
attribute-label .....	35
behaviour-definition-label .....	39
class-label .....	24
group-label .....	37
name-binding-label .....	33
notification-label .....	41
package-label .....	27
parameter-label .....	31
Mots clés relatifs aux modèles ( <i>template keywords</i> )	
ACTION .....	40
ACTION-INFO .....	30,31
ACTION-REPLY .....	30,31
ACTIONS .....	25,27,29,30
ADD .....	27
ADD-REMOVE .....	27
AND ATTRIBUTE IDS .....	41,42
AND SUBCLASSES .....	33
ANY DEFINED BY .....	29
ATTRIBUTE .....	31,32,35

	Page
ATTRIBUTE GROUP.....	37
ATTRIBUTE GROUPS .....	25,27,28
ATTRIBUTES.....	25,27,28,30
BEHAVIOUR.....	25,27,28,31,32,33,34,35,36,37,39,40,41,42
CHARACTERIZED BY.....	24,25,26
CONDITIONAL PACKAGES.....	24,25,26,29
CONTEXT.....	31
CONTEXT SPECIFIC-ERROR .....	37
CREATE.....	30,33,34
DEFAULT VALUE.....	25,27,28
DEFINED AS .....	20,21,39
DELETE .....	30,33,34
DELETES-CONTAINED-OBJECTS .....	33,34
DERIVATION RULE .....	27,28
DERIVED FROM.....	24,25,26,35,36,37
DESCRIPTION .....	37,38
EQUALITY .....	35,36
EVENT-INFO.....	30,31
EVENT-REPLY .....	30,31,32
FIXED.....	37,38
GET.....	27
GET-REPLACE.....	27
GROUP ELEMENTS .....	37,38
INITIAL VALUE .....	25,27,28
MANAGED OBJECT CLASS.....	24
MATCHES FOR.....	35,36,37
MODE CONFIRMED .....	40
NAME BINDING.....	33
NAMED BY SUPERIOR OBJECT CLASS .....	33
NOTIFICATION .....	41
NOTIFICATIONS .....	26,27,29,30
ONLY-IF-NO-CONTAINED-OBJECTS.....	33,34
ORDERING.....	35,36
PACKAGE .....	26,27
PARAMETER.....	31
PARAMETERS .....	35,37,40,41,42
PERMITTED VALUES .....	25,27,28
PRESENT IF .....	24,26

REGISTERED AS .....	20,21,24,26,27,29,31,32,33,34,35,36,37,38,40,41,42
REMOVE.....	27
REPLACE.....	27
REPLACE-WITH-DEFAULT .....	27,28
REQUIRED VALUES .....	25,27,28
SET-COMPARISON.....	35,36
SET-INTERSECTION .....	35,36
SPECIFIC-ERROR.....	30,31,32
SUBORDINATE OBJECT CLASS .....	33,34
SUBSTRINGS.....	35,36
WITH ATTRIBUTE.....	33
WITH ATTRIBUTE SYNTAX.....	35,36,37
WITH INFORMATION SYNTAX.....	40,41,42
WITH REPLY SYNTAX .....	40,41,42
WITH SYNTAX.....	31,32
WITH-AUTOMATIC-INSTANCE-NAMING .....	33,34
WITH-REFERENCE-OBJECT .....	33,34
<b>Outils de notation pour la définition des objets gérés (<i>Notational tools for managed object definition</i>)</b>	
Conventions utilisées dans les définitions de modèle.....	20
Modèle d'action.....	39
Modèle d'attribut.....	35
Modèle d'ensemble .....	26
Modèle de classe d'objets gérés .....	24
Modèle de comportement .....	38
Modèle de corrélation des noms .....	32
Modèle de groupe d'attributs.....	37
Modèle de notification .....	41
Modèle de paramètre .....	29
Vue d'ensemble des outils de notation .....	20
<b>Principes généraux pour la définition des objets gérés (<i>General principles for managed object definition</i>)</b>	
Attributs .....	16
Chevauchements .....	18
Cohérence .....	17
Compteurs.....	18
Compteurs d'événements .....	18
Compteurs d'unités de données de protocole (PDU) .....	18
Compteurs non débordants .....	18
Conservation des informations .....	19

Eléments communs .....	14
Ensembles de valeurs d'attribut.....	16
Hyperclasses sans restrictions.....	15
Identification des objets gérés.....	19
Instanciation des hyperclasses .....	15
Mise à jour des attributs.....	18
Modélisation des SAP.....	17
Notifications.....	19
Objets gérés.....	15
Précision des attributs .....	18
Que gérer?.....	14
Refus de service .....	19
Relations entre valeurs d'attribut.....	16
Statistiques .....	17
Structuration.....	14
Temporisateurs.....	18
Types d'attributs.....	16
Utilisation des opérations.....	19
<b>Questions générales (Global issues)</b>	
Caractéristiques héritées .....	6
Complexité des définitions des objets gérés .....	13
Conformité.....	13
Création et suppression d'objets gérés .....	13
Enregistrement.....	7
Intégrité des relations .....	6
Objets gérés de valeur initiale.....	13
Proposition d'options .....	6
Sources des valeurs initiales d'attribut .....	14
<b>Références des étiquettes de modèle (template label references)</b>	
action-label.....	27,29
attribute-label.....	27,28,32,33,35,36,37,38,41,42
behaviour-definition-label .....	27,28,31,32,33,35,36,40,41,42
class-label.....	24,33
group-label .....	27,28
notification-label .....	27,29
package-label .....	24,26
parameter-label .....	27,28,29,33,34,35,37,40,41,42