



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

X.750

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

(10/96)

SÉRIE X: RÉSEAUX POUR DONNÉES ET
COMMUNICATION ENTRE SYSTÈMES OUVERTS

Gestion OSI

**Technologies de l'information – Interconnexion
des systèmes ouverts – Gestion-systèmes:
Fonction de gestion de la connaissance de
gestion**

Recommandation UIT-T X.750

(Antérieurement «Recommandation du CCITT»)

RECOMMANDATIONS UIT-T DE LA SÉRIE X
RÉSEAUX POUR DONNÉES ET COMMUNICATION ENTRE SYSTÈMES OUVERTS

RÉSEAUX PUBLICS POUR DONNÉES	X.1–X.199
Services et fonctionnalités	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalisation et commutation	X.50–X.89
Aspects réseau	X.90–X.149
Maintenance	X.150–X.179
Dispositions administratives	X.180–X.199
INTERCONNEXION DES SYSTÈMES OUVERTS	X.200–X.299
Modèle et notation	X.200–X.209
Définitions des services	X.210–X.219
Spécifications des protocoles en mode connexion	X.220–X.229
Spécifications des protocoles en mode sans connexion	X.230–X.239
Formulaires PICS	X.240–X.259
Identification des protocoles	X.260–X.269
Protocoles de sécurité	X.270–X.279
Objets gérés de couche	X.280–X.289
Tests de conformité	X.290–X.299
INTERFONCTIONNEMENT DES RÉSEAUX	X.300–X.399
Généralités	X.300–X.349
Systèmes de transmission de données par satellite	X.350–X.399
SYSTÈMES DE MESSAGERIE	X.400–X.499
ANNUAIRE	X.500–X.599
RÉSEAUTAGE OSI ET ASPECTS SYSTÈMES	X.600–X.699
Réseautage	X.600–X.629
Efficacité	X.630–X.649
Dénomination, adressage et enregistrement	X.650–X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680–X.699
GESTION OSI	X.700–X.799
Cadre général et architecture de la gestion-systèmes	X.700–X.709
Service et protocole de communication de gestion	X.710–X.719
Structure de l'information de gestion	X.720–X.729
Fonctions de gestion	X.730–X.799
SÉCURITÉ	X.800–X.849
APPLICATIONS OSI	X.850–X.899
Engagement, concomitance et rétablissement	X.850–X.859
Traitement transactionnel	X.860–X.879
Opérations distantes	X.880–X.899
TRAITEMENT OUVERT RÉPARTI	X.900–X.999

Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'UIT. Au sein de l'UIT-T, qui est l'entité qui établit les normes mondiales (Recommandations) sur les télécommunications, participent quelque 179 pays membres, 84 exploitations de télécommunications reconnues, 145 organisations scientifiques et industrielles et 38 organisations internationales.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la Conférence mondiale de normalisation des télécommunications (CMNT), (Helsinki, 1993). De plus, la CMNT, 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 de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI. Le texte de la Recommandation X.750 de l'UIT-T a été approuvé le 5 octobre 1996. Son texte est publié, sous forme identique, comme Norme internationale ISO/CEI 10164-16.

NOTE

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

© UIT 1997

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'UIT.

TABLE DES MATIÈRES

	<i>Page</i>
1	Domaine d'application..... 1
2	Références normatives 1
2.1	Recommandations Normes internationales identiques..... 2
2.2	Paires de Recommandations Normes internationales équivalentes par leur contenu technique 3
2.3	Autres références 3
3	Définitions..... 4
3.1	Définitions du modèle de référence de base 4
3.2	Définitions de la structure de la couche Application 4
3.3	Définitions des tests de conformité de l'OSI 4
3.4	Définitions de l'Annuaire 4
3.5	Définitions du cadre de gestion..... 5
3.6	Définitions de l'aperçu général de la gestion-systèmes 5
3.7	Définitions du service CMIS..... 5
3.8	Définitions du modèle d'informations de gestion..... 6
3.9	Définitions des directives pour la définition des objets gérés 6
3.10	Définitions des spécifications et directives pour l'établissement des formulaires de déclaration de conformité d'instances associés à la gestion OSI 6
3.11	Définitions du modèle général de relation 7
3.12	Autre définition..... 7
4	Abréviations 7
5	Conventions..... 8
6	Exigences 8
6.1	Connaissance des classes d'objets gérés..... 9
6.2	Connaissance des instances de classes d'objets gérés 9
6.3	Connaissance des relations..... 9
6.4	Connaissance de l'utilisateur du service MIS..... 9
6.5	Connaissance des définitions d'informations de gestion 10
7	Modèle..... 10
7.1	Objets gérés de connaissance de gestion..... 10
7.2	Objets d'Annuaire de connaissance de gestion 12
7.3	Relations entre les exigences relatives à la connaissance de gestion et le modèle..... 18
8	Définitions génériques 19
8.1	Objets gérés..... 19
8.2	Relations de dénomination..... 23
8.3	Actions 24
8.4	Compatibilité..... 25
8.5	Définitions génériques provenant de la fonction de gestion d'objets 25
8.6	Objets d'Annuaire 26
9	Définition du service 28
9.1	Introduction..... 28
9.2	Objets gérés de connaissance de gestion..... 28
9.3	Services de notification..... 28
9.4	Services d'action..... 29
9.5	Services d'Annuaire 29
10	Unités fonctionnelles de gestion-systèmes 30

	<i>Page</i>	
11	Protocole et syntaxe abstraite	31
11.1	Syntaxe abstraite	31
11.2	Éléments de procédure pour l'obtention de représentation textuelle	35
11.3	Éléments de procédure pour la recherche d'arbre d'informations de gestion	37
11.4	Négociation des unités fonctionnelles	37
11.5	Éléments caractéristiques du protocole CMIP	37
12	Relations avec d'autres fonctions et avec l'Annuaire	37
13	Conformité	38
13.1	Conformité statique	38
13.2	Conformité dynamique	38
13.3	Exigences relatives aux déclarations de conformité d'une implémentation de la gestion	38
Annexe A	– Objets gérés de connaissance de gestion	39
A.1	Allocation of object identifiers	39
A.2	Repertoire managed object definitions	39
A.3	Definition managed object definitions	42
A.4	Discovery managed object definitions	52
Annexe B	– Objets d'Annuaire de connaissance de gestion	54
B.1	Repertoire Directory object definitions	54
B.2	Definition Directory object definitions	55
Annexe C	– Formulaire MCS	61
C.1	Introduction	61
C.2	Identification of the implementation	62
C.3	Identification of the documents in which the management information is defined	63
C.4	Management conformance summary	63
Annexe D	– Formulaire MICS	68
D.1	Introduction	68
D.2	Instructions for completing the MICS proforma to produce an MICS	68
D.3	Statement of conformance to the management information	68
Annexe E	– Formulaire MOCS	72
E.1	Introduction	72
E.2	Instructions for completing the MOCS proforma to produce an MOCS	72
E.3	Statements of conformance to the managed object classes	72
Annexe F	– Formulaire MRCS	102
F.1	Introduction	102
F.2	Instructions for completing the MRCS proforma for name binding to produce an MRCS	102
F.3	Statement of conformance to the name binding	102
Annexe G	– Directives pour l'utilisation des objets de connaissance de gestion	104
Annexe H	– Options d'accès pour l'arbre d'annuaire de connaissance des définitions	105

Résumé

Un système d'exploitation (OS, *operations systems*) ne peut être exploité en interfonctionnement avec un autre OS ou avec un élément de réseau (NE, *network element*) que s'il connaît les types précis de messages de gestion et d'informations de gestion que ce système OS ou cet élément NE peut prendre en charge pour les interactions de gestion. Il existe de nombreuses façons d'obtenir ces connaissances et de les incorporer dans un logiciel de système d'exploitation ou d'élément de réseau. La présente Recommandation indique les moyens de spécifier les informations et les messages de gestion des connaissances, qui permettent de transférer en ligne d'un système à un autre les détails précis afférents à ces informations. Par ailleurs, l'administration d'un réseau de gestion sera rendue plus efficace par l'utilisation de systèmes dotés de cette capacité de connaissances de gestion; en effet, le transfert de ces connaissances est assuré par l'utilisation de messages normalisés de gestion des connaissances conformes à la Recommandation X.700, si bien que les constructeurs qui ont besoin que leurs systèmes soient exploités en interfonctionnement ne sont pas tenus d'obtenir un accord local pour les messages fondés sur la gestion des systèmes OSI; durant l'exploitation, il est possible de vérifier la capacité d'un OS ou d'un NE et il existe un moyen normalisé d'augmenter le volume des informations et des messages de gestion que ce système ou cet élément de réseau peut prendre en charge. Au lieu d'utiliser les messages X.700, on peut aussi utiliser les services de l'Annuaire X.500 et pour ce faire, la présente Recommandation | Norme internationale définit les entrées de l'Annuaire regroupant ces informations.

NORME INTERNATIONALE

RECOMMANDATION UIT-T

**TECHNOLOGIES DE L'INFORMATION – INTERCONNEXION DES
SYSTÈMES OUVERTS – GESTION-SYSTÈMES: FONCTION DE
GESTION DE LA CONNAISSANCE DE GESTION**

1 Domaine d'application

La présente Recommandation | Norme internationale définit la fonction de gestion de la connaissance de gestion. Cette fonction appartient à la classe des fonctions de gestion-systèmes. Elle peut être utilisée par un processus d'application, dans un environnement de gestion centralisé ou réparti, pour participer à la gestion-systèmes telle qu'elle est définie par le cadre général de gestion OSI décrit dans la Rec. X.700 du CCITT et l'ISO/CEI 7498-4. La fonction définie dans la présente Recommandation | Norme internationale est composée de définitions génériques et de services. Elle se situe dans la couche Application du modèle OSI (Rec. UIT-T X.200 | ISO/CEI 7498-1). Sa définition respecte le modèle décrit dans la Rec. UIT-T X.207 | ISO/CEI 9545. Le rôle des fonctions de gestion-systèmes est précisé dans la Rec. X.701 du CCITT | ISO/CEI 10040.

La présente Recommandation | Norme internationale:

- identifie l'ensemble des exigences de l'utilisateur auxquelles satisfait la fonction;
- fournit un modèle de comportement pour les objets de connaissance de gestion;
- spécifie les exigences de la fonction relatives à la gestion ainsi que les moyens d'y satisfaire par la spécification d'objets de gestion et de leur comportement ou par la spécification d'objets d'Annuaire;
- spécifie l'application de ces services sur ceux du service CMIS;
- spécifie la syntaxe abstraite des paramètres des unités MAPDU qui serviront pour faire référence aux objets gérés et leurs caractéristiques.

La présente Recommandation | Norme internationale:

- ne définit la nature d'aucune implémentation qui serait destinée à offrir la fonction de gestion de la connaissance de gestion;
- ne spécifie pas de quelle manière l'utilisateur de la fonction de la connaissance de gestion doit exercer la gestion;
- ne définit la nature d'aucune des interactions entraînant l'emploi de la fonction de gestion de la connaissance de gestion;
- ne spécifie pas les services nécessaires à l'établissement et la libération, normale ou anormale, d'une association de gestion;
- ne définit pas les interactions qui résultent de l'emploi concomitant de plusieurs fonctions de gestion;
- ne définit pas les exigences que l'usage de ces fonctions ou la poursuite d'activités connexes font porter sur l'établissement de connexions ou sur les autorisations;
- n'empêche pas la définition ultérieure d'autres classes d'objets de connaissance de gestion.

2 Références normatives

Les Recommandations et 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. Toutes Recommandations et Normes sont sujettes 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 et Normes indiquées ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur. Le Bureau de la normalisation des télécommunications de l'UIT tient à jour une liste des Recommandations de l'UIT-T en vigueur.

2.1 Recommandations | Normes internationales identiques

- Recommandation UIT-T X.200 (1994) | ISO/CEI 7498-1:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Modèle de référence de base: le modèle de référence de base.*
- Recommandation UIT-T X.207 (1993) | ISO/CEI 9545:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure de la couche application.*
- Recommandation UIT-T X.210 (1993) | ISO/CEI 10731:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Modèle de référence de base: conventions pour la définition des services de l'interconnexion de systèmes ouverts.*
- Recommandation UIT-T X.217 (1995) | ISO/CEI 8649:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Définition de service applicable à l'élément de service de contrôle d'association.*
- Recommandation UIT-T X.501 (1993) | ISO/CEI 9594-2:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: les modèles.*
- Recommandation UIT-T X.511 (1993) | ISO/CEI 9594-3:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: définition du service abstrait.*
- Recommandation UIT-T X.520 (1993) | ISO/CEI 9594-6:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: types d'attributs sélectionnés.*
- Recommandation UIT-T X.521 (1993) | ISO/CEI 9594-7:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: classes d'objets sélectionnées.*
- Recommandation UIT-T X.680 (1994) | ISO/CEI 8824-1:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base.*
- Recommandation UIT-T X.681 (1994) | ISO/CEI 8824-2:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des objets informationnels.*
- Recommandation UIT-T X.682 (1994) | ISO/CEI 8824-3:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des contraintes.*
- Recommandation UIT-T X.683 (1994) | ISO/CEI 8824-4:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: paramétrage des spécifications de la notation de syntaxe abstraite numéro un.*
- Recommandation UIT-T X.690 (1994) | ISO/CEI 8825-1:1995, *Technologies de l'information – Règles de codage de la notation de syntaxe abstraite numéro un: spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives.*
- Recommandation X.701 du CCITT (1992) | ISO/CEI 10040:1992, *Technologies de l'information – Interconnexion des systèmes ouverts – Aperçu général de la gestion-systèmes.*
- Recommandation X.720 du CCITT (1992) | ISO/CEI 10165-1:1993, *Technologies de l'information – Interconnexion des 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 des systèmes ouverts – Structure des informations de gestion: définition des informations de gestion.*
- Recommandation X.722 du CCITT (1992) | ISO/CEI 10165-4:1992, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure des informations de gestion: directives pour la définition des objets gérés.*
- Recommandation UIT-T X.724 (1993) | ISO/CEI 10165-6:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure de l'information de gestion: spécifications et directives pour l'établissement des formulaires de déclaration de conformité d'implémentations associés à la gestion OSI.*
- Recommandation UIT-T X.725 (1995) | ISO/CEI 10165-7:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Structure de l'information de gestion: modèle général de relation.*

- Recommandation X.730 du CCITT (1992) | ISO/CEI 10164-1:1993, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de gestion des objets.*
- Recommandation X.735 du CCITT (1992) | ISO/CEI 10164-6:1993, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: fonction de commande des registres de consignation.*
- Recommandation UIT-T X.741 (1995) | ISO/CEI 10164-9:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – Gestion-systèmes: objets et attributs de contrôle d'accès.*

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

- 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 1 (ASN.1)* [Publiée actuellement en anglais seulement et édition retenue à titre provisoire].
- Recommandation X.209 du CCITT (1988), *Spécification des règles de codage de base pour la notation de syntaxe abstraite numéro un (ASN.1).*
ISO/CEI 8825:1990, *Technologies de l'information – Interconnexion de systèmes ouverts – Spécification de règles de base pour coder la notation de syntaxe abstraite numéro une (ASN.1)* [Publiée actuellement en anglais seulement et édition retenue à titre provisoire].
- Recommandation X.290 du CCITT (1992), *Cadre général et méthodologie des tests de conformité OSI pour les Recommandations sur les protocoles pour les applications du CCITT – Concepts généraux.*
ISO/CEI 9646-1:1994, *Technologies de l'information – Interconnexion de systèmes ouverts – Cadre général et méthodologie des tests de conformité OSI – Partie 1: Concepts généraux* (Publiée actuellement en anglais seulement).
- Recommandation X.291 du CCITT (1992), *Cadre général et méthodologie des tests de conformité OSI pour les Recommandations sur les protocoles pour les applications du CCITT – Spécification des suites de tests abstraites.*
ISO/CEI 9646-2:1994, *Technologies de l'information – Interconnexion de systèmes ouverts – Cadre général et méthodologie des tests de conformité OSI – Partie 2: Spécification des suites de tests abstraites* (Publiée actuellement en anglais seulement).
- Recommandation UIT-T X.296 (1995), *Cadre général et méthodologie des tests de conformité OSI pour les Recommandations sur les protocoles pour les applications de l'UIT-T – Déclarations de conformité d'instance.*
ISO/CEI 9646-7:1995, *Technologies de l'information – Interconnexion de systèmes ouverts – Essais de conformité – Méthodologie générale et procédures – Partie 7: Déclarations de conformité des mises en œuvre* (Publiée actuellement en anglais seulement).
- Recommandation X.700 du CCITT (1992), *Cadre de gestion pour l'interconnexion de systèmes ouverts 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.*

2.3 Autres références

- ISO/CEI ISP 11183-1:1992, *Technologies de l'information – Profil normalisé international AOMIn pour la gestion OSI – Communications d'informations de gestion – Partie 1: Spécification des protocoles d'ACSE, de présentation et de session utilisés par ROSE et CMISE.*

- ISO/CEI ISP 11183-2:1992, *Technologies de l'information – Profil normalisé international AOM1n pour la gestion OSI – Communications d'informations de gestion – Partie 2: CMISE/ROSE pour AOM12 – Communications étendues d'informations de gestion.*
- ISO/CEI ISP 11183-3:1992, *Technologies de l'information – Profil normalisé international AOM1n pour la gestion OSI – Communications d'informations de gestion – Partie 3: CMISE/ROSE pour AOM11 – Communications de base d'information de gestion.*

3 Définitions

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

3.1 Définitions du modèle de référence de base

La présente Recommandation | Norme internationale utilise les termes suivants qui sont définis dans la Rec. UIT-T X.200 | ISO/CEI 7498-1:

- a) système ouvert;
- b) unité de données de protocole (N);
- c) gestion-systèmes;
- d) entité d'application de gestion-systèmes.

3.2 Définitions de la structure de la couche Application

La présente Recommandation | Norme internationale utilise le terme suivant qui est défini dans la Rec. UIT-T X.207 | ISO/CEI 9545:

- élément de service d'application.

3.3 Définitions des tests de conformité de l'OSI

La présente Recommandation | Norme internationale utilise les termes suivants qui sont définis dans la Rec. X.290 du CCITT et l'ISO/CEI 9646-1 et dans la Rec. UIT-T X.296 et l'ISO/CEI 9646-7:

- a) formulaire ICS;
- b) déclaration de conformité d'une réalisation;
- c) formulaire PICS;
- d) déclaration de conformité d'une réalisation de protocole.

3.4 Définitions de l'Annuaire

La présente Recommandation | Norme internationale utilise les termes suivants définis dans la Rec. UIT-T X.501 | ISO/CEI 9594-2:

- a) entrée pseudonyme;
- b) attribut;
- c) classe d'objets auxiliaire;
- d) entrée d'Annuaire;
- e) arbre d'informations de l'Annuaire;
- f) nom d'Annuaire;
- g) classe d'objets d'Annuaire;
- h) agent de système d'Annuaire;
- i) agent d'utilisateur d'Annuaire;
- j) nom distinctif;
- k) règle de structure du DIT;
- l) nom distinctif relatif;

- m) classe d'objets structurelle;
- n) subordonné;
- o) supérieur;
- p) L'Annuaire.

NOTE – Les termes et expressions «attribut», «nom distinctif», «nom distinctif relatif», «subordonné» et «supérieur» sont définis à la fois dans la Rec. UIT-T X.501 | ISO/CEI 9594-2 ainsi que, soit dans la Rec. X.710 du CCITT et l'ISO/CEI 9595, soit dans la Rec. X.720 du CCITT | ISO/CEI 10165-1 par suite de ressemblances entre le modèle de l'Annuaire et le modèle d'informations de gestion. L'usage qui est fait de ces termes et expressions par les deux modèles n'est cependant pas totalement identique. Le contexte dans lequel ils sont employés permet de déterminer la définition qui convient.

3.5 Définitions du cadre de gestion

La présente Recommandation | Norme internationale utilise le terme suivant défini dans la Rec. X.700 du CCITT et l'ISO/CEI 7498-4:

- objet géré.

3.6 Définitions de l'aperçu général de la gestion-systèmes

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

- a) agent;
- b) rôle d'agent;
- c) définitions génériques;
- d) système géré;
- e) déclaration de conformité d'objet géré;
- f) informations de gestion;
- g) déclaration de conformité d'informations de gestion;
- h) opération de gestion;
- i) gestionnaire;
- j) rôle de gestionnaire;
- k) système gestionnaire;
- l) formulaire MICS;
- m) utilisateur du service MIS;
- n) formulaire MOCS;
- o) notification;
- p) type de notification;
- q) élément de service d'application de gestion-systèmes;
- r) unité fonctionnelle de gestion-systèmes.

3.7 Définitions du service CMIS

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

- a) attribut;
- b) élément de service commun d'informations de gestion;
- c) service commun d'informations de gestion;
- d) (attribut) valué par un ensemble.

3.8 Définitions du modèle d'informations 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) classe allomorphique;
- d) groupe d'attributs;
- e) identificateur d'attribut;
- f) type d'attribut;
- g) comportement;
- h) caractéristique;
- i) bloc de propriétés conditionnel;
- j) confinement;
- k) nom distinctif;
- l) instanciation;
- m) bloc de propriétés obligatoire;
- n) relation de dénomination;
- o) arbre de dénomination;
- p) bloc de propriétés;
- q) paramètre;
- r) nom distinctif relatif;
- s) spécialisation;
- t) sous-classe;
- u) hyperclasse;
- v) (objet) subordonné;
- w) (objet) supérieur.

3.9 Définitions des directives pour la définition des objets gérés

La présente Recommandation | Norme internationale utilise les termes suivants définis dans la Rec. X.722 du CCITT | ISO/CEI 10165-4:

- a) définition de classe d'objets gérés;
- b) modèle.

3.10 Définitions des spécifications et directives pour l'établissement des formulaires de déclaration de conformité d'instances associés à la gestion OSI

La présente Recommandation | Norme internationale utilise les termes suivants définis dans la Rec. UIT-T X.724 | ISO/CEI 10165-6:

- a) déclaration de conformité de relation gérée;
- b) récapitulatif de conformité de gestion;
- c) formulaire MCS;
- d) formulaire MRCS.

3.11 Définitions du modèle général de relation

La présente Recommandation | Norme internationale utilise le terme suivant défini dans la Rec. UIT-T X.725 | ISO/CEI 10165-7:

- relation gérée.

3.12 Autre définition

3.12.1 objet de connaissance de gestion: objet de gestion ou objet d'Annuaire qui rend disponible la connaissance de gestion.

4 Abréviations

Pour les besoins de la présente Recommandation | Norme internationale, les abréviations suivantes sont utilisées:

ACSE	Elément de service de contrôle d'association (<i>association control service element</i>)
ASE	Elément de service d'application (<i>application service element</i>)
ASN.1	Notation de syntaxe abstraite numéro un (<i>abstract syntax notation one</i>)
CMIP	Protocole commun d'informations de gestion (<i>common management information protocol</i>)
CMIS	Service commun d'informations de gestion (<i>common management information service</i>)
CMISE	Elément de service commun d'informations de gestion (<i>common management information service element</i>)
conf.	Confirmation
dem.	Demande
DIT	Arbre d'informations d'annuaire (<i>directory information tree</i>)
DKDT	Arbre d'annuaire de connaissance de définitions (<i>definition knowledge directory tree</i>)
DSA	Agent de système d'annuaire (<i>directory system agent</i>)
DUA	Agent d'utilisateur d'annuaire (<i>directory user agent</i>)
GDMO	Directives pour la définition des objets gérés (<i>guidelines for the definition of managed objects</i>)
ICS	Déclaration de conformité d'implémentation (<i>implementation conformance statement</i>)
ID	Identificateur (<i>identifier</i>)
ind.	Indication
ISP	Profil de norme internationale (<i>international standardized profile</i>)
MAPDU	Unité de données de protocole d'application de gestion (<i>management application protocol data unit</i>)
MCS	Récapitulatif de conformité de gestion (<i>management conformance summary</i>)
MICS	Déclaration de conformité d'informations de gestion (<i>management information conformance statement</i>)
MIS	Service d'informations de gestion (<i>management information service</i>)
MKM	Fonction de gestion de la connaissance de gestion (<i>management knowledge management function</i>)
MOCS	Déclaration de conformité d'objet géré (<i>managed object conformance statement</i>)
MRCS	Déclaration de conformité de relation gérée (<i>managed relationship conformance statement</i>)
PDU	Unité de données de protocole (<i>protocol data unit</i>)
PICS	Déclaration de conformité d'implémentation de protocole (<i>protocol implementation conformance statement</i>)

RDN	Nom distinctif relatif (<i>relative distinguished name</i>)
rép.	Réponse
SMAE	Entité d'application de gestion-systèmes (<i>systems management application entity</i>)
SMAPM	Machine protocolaire d'application de gestion-systèmes (<i>systems management application protocol machine</i>)
SMASE	Élément de service d'application de gestion-systèmes (<i>systems management application service element</i>)

5 Conventions

La présente Recommandation | Norme internationale définit les services offerts par la fonction de gestion de connaissance de gestion en respectant les conventions de description définies dans la Rec. UIT-T X.210 | ISO/CEI 10731.

Les notations suivantes sont utilisées dans les tableaux de paramètres des primitives de service:

M	Le paramètre est obligatoire (<i>mandatory</i>)
C	Le paramètre est conditionnel (<i>conditional</i>)
(=)	La valeur du paramètre est identique à celle du paramètre correspondant dans l'interaction décrite par la primitive de service apparentée précédente
U	L'usage du paramètre est facultatif, au gré de l'utilisateur de service (<i>service-user option</i>)
–	Le paramètre n'est pas présent dans l'interaction décrite par la primitive de service en question
P	Le paramètre fait l'objet des contraintes imposées par la Rec. UIT-T X.710 et l'ISO/CEI 9595.

NOTE – Les paramètres notés «P» dans les tableaux de service de la présente Recommandation | Norme internationale sont appliqués directement sur les paramètres correspondants de la primitive de service du CMIS, sans changement ni de la valeur sémantique ni de la syntaxe du paramètre. Les autres paramètres servent à construire une unité MAPDU.

6 Exigences

Les systèmes ouverts qui participent à des associations de gestion-systèmes OSI, ou qui sont susceptibles d'y participer, ont besoin d'un savoir particulier pour connaître les systèmes ouverts auxquels ils s'associent, pour rendre l'association possible et pour assurer les fonctions de la gestion-systèmes OSI. Les exigences spécifiques sont présentées ci-dessous, classées par domaine de connaissance.

On reconnaît trois types de connaissance de gestion:

- *connaissance des répertoires*, donnant des informations sur ce que le système géré est capable de faire. Il existe trois catégories de capacités:
 - capacités relatives aux classes d'objets gérés;
 - capacités relatives aux classes de relations gérées;
 - capacités relatives aux fonctions.

Il est nécessaire que le gestionnaire soit capable de découvrir les capacités des systèmes gérés sans avoir à accéder à chacun d'entre eux.

- *connaissance des définitions*, donnant des informations sur la spécification formelle des classes d'objets gérés, les relations de dénomination, etc. Il s'agit, par exemple, des modèles de classes, de relations de dénomination.
- *connaissance des instances*, donnant des informations précisant quels objets gérés et quelles relations gérées le système géré rend visibles et quelles entités d'application de gestion-systèmes sont utilisables pour établir une association avec un système géré qui rend visible un objet géré donné.

Les mécanismes de partage de la connaissance doivent assurer le respect des exigences de contrôle d'accès touchant au système ouvert au sujet duquel de l'information est fournie. En particulier, les mécanismes de partage servant à la gestion-systèmes OSI ne doivent pas révéler à un utilisateur des informations qu'il lui serait par ailleurs interdit d'obtenir des services de gestion-systèmes OSI.

Les exigences à satisfaire quant à l'acquisition des connaissances du répertoire, des définitions et des instances sont subdivisées en cinq catégories majeures (voir Tableau 1).

Tableau 1 – Exigences

Numéro de paragraphe	Catégories principales	Types de connaissance de gestion		
		Connaissance des répertoires	Connaissance des définitions	Connaissance des instances
6.1	Connaissance des classes d'objets gérés	X		
6.2	Connaissance des instances de classes d'objets gérés			X
6.3	Connaissance des relations	X		X
6.4	Connaissance de l'utilisateur du service MIS	X		X
6.5	Connaissance des définitions d'informations de gestion		X	

6.1 Connaissance des classes d'objets gérés

Il faut pouvoir déterminer dans le cadre de la connaissance des répertoires:

- les classes d'objets gérés qu'un système géré donné peut rendre disponibles;
- les contraintes appliquées aux objets gérés d'une classe donnée (par exemple, l'ensemble des valeurs implémentées d'un attribut donné);
- pour une classe d'objets donnée, les blocs de propriétés conditionnels qui sont toujours inclus dans une instance de la classe et ceux qui ne le sont jamais;
- les valeurs initiales d'une instance d'une classe particulière qui peut être créée dans un système géré donné;
- les allomorphismes pris en compte pour chacune des classes d'objets gérés mises en œuvre.

6.2 Connaissance des instances de classes d'objets gérés

Sans qu'il soit obligatoire de faire appel à l'unité fonctionnelle de sélection d'objets multiples du service CMIS, il faut pouvoir déterminer dans le cadre de la connaissance des instances:

- les instances d'une classe donnée d'objets gérés qu'un système géré donné rend visibles;
- les systèmes gérés qui rendent visibles un objet géré donné;
- le nombre d'instances d'une classe donnée d'objets gérés qu'un système géré donné rend visible.

6.3 Connaissance des relations

Il faut pouvoir déterminer dans le cadre de la connaissance des répertoires:

- les classes de relations mises en œuvre par un système géré donné;
- les relations entre rôles, en particulier les relations de dénomination, qui peuvent exister entre les classes d'objets gérés rendues visibles par un système géré donné.

Il faut aussi pouvoir déterminer dans le cadre de la connaissance des instances:

- les instances de relations qui existent entre les objets gérés rendus visibles par un système géré donné ou par différents systèmes gérés.

6.4 Connaissance de l'utilisateur du service MIS

Il faut pouvoir déterminer dans le cadre de la connaissance des répertoires:

- les titres des entités SMAE des systèmes gérés ou gestionnaires avec lesquels peuvent être établies des associations de gestion;
- les adresses de présentation de ces SMAE;

ISO/CEI 10164-16 : 1997 (F)

- les contextes d'application mis en œuvre par ces SMAE, y compris les syntaxes et les identificateurs d'informations associés à ces contextes d'application ;
- quelles unités fonctionnelles de SMASE, de CMISE, de l'ACSE ou d'autres éléments de service d'application utilisés dans la gestion, sont mises en œuvre par ces SMAE pour leurs associations de gestion-systèmes;
- quels profils normalisés sont mis en œuvre.

Il faut aussi pouvoir déterminer dans le cadre de la connaissance des instances:

- étant donné un nom global d'objet géré, quelles entités de gestion-systèmes peuvent être utilisées afin d'établir une association avec un système géré qui rend visible l'objet en question.

6.5 Connaissance des définitions d'informations de gestion

Il faut pouvoir déterminer dans le cadre de la connaissance des définitions:

- les définitions formelles des informations de gestion, comprenant les classes d'objets gérés, les relations de dénomination, les catégories de tests, les classes de relations et toute information afférente;

NOTE – Lorsqu'un service d'Annuaire est disponible, tout ou partie de ces informations peut y être contenu et en être obtenu. L'Annexe B donne les définitions qui conviennent dans le cas d'une telle solution.

- quelles définitions d'informations de gestion sont compréhensibles par un système gestionnaire donné.

7 Modèle

Ce modèle explique comment rendre disponibles les divers types de connaissance de gestion. Ces informations sont spécifiées d'abord sous:

- 1) la forme d'objets gérés, selon la Rec. X.722 du CCITT | ISO/CEI 10165-4 (directives pour la définition des objets gérés);
- 2) la forme d'objets d'Annuaire, selon la Rec. UIT-T X.501 | ISO/CEI 9594-2 (l'Annuaire: modèles).

Ces objets de connaissance de gestion sont respectivement décrits aux 7.1 et 7.2.

Ce modèle est cohérent avec la manière dont la normalisation de la connaissance de gestion est spécifiée dans la Rec. X.701 du CCITT | ISO/CEI 10040.

7.1 Objets gérés de connaissance de gestion

Le modèle applicable aux objets gérés utilisés pour enregistrer et partager certains aspects de la connaissance de gestion est présenté dans les paragraphes suivants.

7.1.1 Objets gérés répertoires

Les objets gérés répertoires satisfont aux exigences concernant la détermination des informations indiquant ce qu'un système géré est capable de faire.

7.1.1.1 Objets gérés répertoires

Les objets gérés répertoires (*repertoire managed objects*) satisfont aux exigences concernant l'identification des classes d'objets gérés, des relations de dénomination, des classes de relations gérées et des profils CMIP qu'un système géré met en œuvre.

Les objets gérés de la classe d'objets gérés répertoires rendent disponibles les listes des classes d'objets gérés mises en œuvre par le système géré. Chaque liste comprend la classe réelle et les classes allomorphiques. Sont aussi rendues disponibles une liste de relations de dénomination et une liste de classes de relations que le système géré met en œuvre. Une instance de l'objet géré répertoire représentant les composantes du système géré chargées de la communication contient aussi un attribut pour identifier les profils du CMIP que le système géré met en œuvre.

L'emploi d'un nom distinctif relatif figé pour dénommer un objet géré répertoire permet d'accéder de manière uniforme à cet objet. C'est pourquoi il est fourni une relation de dénomination pour pouvoir assurer le confinement d'un objet géré répertoire à l'intérieur d'un objet géré système à l'aide d'un nom distinctif relatif figé. Dès lors qu'un système géré contient un objet géré répertoire quelconque, il doit en contenir un dont le nom provient de système avec la valeur de RDN «REP1». L'information de répertoire peut aussi être rendue disponible par d'autres moyens, par emploi, par exemple, de l'Annuaire.

Une instance de la classe d'objets gérés répertoires peut contenir comme subordonnés des objets gérés répertoires de classe d'objets gérés. Chacun de ces objets rend disponible la connaissance des blocs de propriétés pris en charge dans le cadre d'une classe d'objets gérés particulière.

7.1.1.2 Objets gérés répertoires de classe d'objets gérés

Les objets gérés répertoires de classe d'objets gérés (*managed object class repertoire managed objects*) et leurs sous-classes identifient les informations relatives à l'implémentation d'une classe d'objets gérés. Ils peuvent être dénommés par l'attribut "identificateur de répertoire de classe d'objets gérés". Celui-ci prend la valeur de l'identificateur d'objet de la classe dont sont représentées les informations d'implémentation. Ils peuvent être confinés à l'intérieur d'un objet géré répertoire.

Un objet géré répertoire de classe d'objets gérés possède un attribut valué par un ensemble afin d'identifier les blocs de propriétés conditionnels pris en compte par l'implémentation.

Un objet géré répertoire de classe d'objets gérés peut posséder un attribut valué par un ensemble afin d'identifier les valeurs prises en compte par l'implémentation, telles que valeurs initiales, domaine de valeurs et valeurs par défaut.

7.1.2 Objets gérés définitions

Les objets gérés définitions (*definition managed objects*) satisfont aux exigences concernant la détermination des définitions d'informations de gestion.

La connaissance des définitions est nécessaire au cours de l'association. Certains systèmes gestionnaires ne pourront faire qu'un usage limité de la connaissance des définitions acquise durant l'association. Pour des systèmes gestionnaires simples, par exemple, les classes d'objets gérés traitées par un système géré ne seront utiles que s'ils ont déjà eux-mêmes mis en œuvre ces mêmes classes, ou des classes compatibles.

D'autres systèmes gestionnaires auront de plus grandes possibilités, étant capables d'interpréter les définitions obtenues durant l'association. Leurs utilisateurs disposeront alors des moyens de manipuler des objets gérés appartenant à des classes jusqu'alors inconnues. Par exemple, un système gestionnaire pourrait avoir les moyens de demander à un système géré quelles actions sont prises en charge par une classe donnée d'objets gérés. En ce qui concerne les actions qu'il connaît déjà, le système questionneur les a déjà mises en œuvre. Pour les actions inconnues, il peut être possible d'obtenir plus d'informations, comme la syntaxe des demandes.

Il est inévitable que de telles interrogations conduisent à l'acquisition d'informations qui ne peuvent être utilisées hors de la connaissance de leur valeur sémantique. Mais il n'existe pas de raisons de principe qui interdiraient de mettre des descriptions d'objets gérés à la disposition d'utilisateurs intelligents. Il n'existe pas non plus de raisons de principe qui empêcheraient ces utilisateurs de faire usage de ces informations. Un cas d'application est celui du rapport d'évènement: bien qu'il ne puisse pas être attendu d'un système gestionnaire, qui n'a jusque là pas connu l'existence d'une notification particulière, qu'il agisse de manière spécifique à la réception d'une telle notification, le besoin peut exister de porter autant d'informations que possible sur cette notification à la connaissance des utilisateurs. Il faudrait à cette fin émettre des demandes vers le système géré émetteur pour obtenir la spécification GDMO de la notification et la syntaxe de son argument. Le décodage de la notification et la présentation de son argument en seraient facilités.

Il existe deux approches pour la spécification d'objets gérés définitions. Ou bien l'objet place les spécifications dans un attribut de forme textuelle; ou bien il dispose d'attributs qui représentent les éléments de spécification. La première méthode est très simple. Elle rend aisé le partage des informations d'un même format qui peuvent provenir de n'importe quelle origine textuelle. La deuxième méthode donne à la gestion OSI l'accès à toutes les caractéristiques de la spécification.

L'étude des besoins des utilisateurs de ces informations permet de déterminer le niveau de détail auquel devrait se faire le partage de la connaissance des définitions. Les utilisateurs se divisent en deux catégories: ceux qui désirent parcourir les informations afin, par exemple, de déterminer les attributs d'une classe d'objets gérés particulière et ceux qui désirent les interpréter, comme dans le cas d'un compilateur de syntaxe ASN.1 ou de directives GDMO. Ces derniers se serviront en général du texte comme information d'entrée. Les premiers peuvent se contenter de récupérer les informations sous leur forme textuelle pour les explorer à loisir. Certains d'entre eux pourtant peuvent désirer avoir accès à des éléments de connaissance spécifique, afin de déterminer s'il est possible et intéressant d'essayer de traiter des objets gérés appartenant à des classes jusque là inconnues. Des informations de cette nature peuvent être nécessaires par exemple pour la configuration de discriminateurs d'émission d'évènements.

Il existe trois types d'objets gérés définitions: documents (*documents*), modèles (*templates*) et modules ASN.1 (*ASN.1 modules*). Les objets gérés documents représentent une collection de modèles et de modules ASN.1. Ils comportent des attributs qui identifient le nom et l'identificateur du document, en accord avec les spécifications des GDMO, et les spécifications GDMO et ASN.1 sous forme textuelle. Les objets gérés modèles sont confinés dans les objets gérés documents. Ils sont dénommés par l'étiquette du modèle GDMO et comportent des attributs qui représentent les groupes constitutifs du modèle. Les objets gérés modules ASN.1 sont confinés dans les objets gérés documents. Ils sont dénommés par la référence du module ASN.1 et comportent des attributs qui représentent, le cas échéant, l'identificateur d'objet, la version d'ASN.1 et le module ASN.1 sous forme textuelle.

Deux relations de dénomination sont fournies pour assurer le confinement d'un objet géré document à l'intérieur d'un objet géré système. Ces relations n'empêchent pas que soient établies d'autres relations de dénomination, permettant l'accès aux informations de définition par usage de l'Annuaire, ou via des tierces parties par emploi des services de gestion-systèmes.

7.1.3 Objets gérés explorations

Les objets gérés explorations (*discovery managed objects*) satisfont aux exigences concernant la détermination des objets gérés rendus visibles par un système géré.

La découverte des objets gérés qu'un système géré rend visibles est une nécessité, indépendamment du niveau de capacité du protocole CMIP mis en œuvre. La conformité au CMIP n'exige pas l'implémentation de capacités de recherche. Il n'est donc pas possible de rechercher les objets gérés rendus visibles par le système géré. Ce besoin est satisfait par le concept d'objet géré exploration.

L'objet géré exploration agit par recherche en arbre. L'argument d'action contient l'objet de base, le domaine d'application désiré pour la recherche et un paramètre facultatif de demande de classe. La réponse se compose d'un groupe de structures arborescentes qui, collectivement, couvrent le domaine demandé en dessous de l'objet de base demandé. Chaque arbre est doté du nom distinctif de sa racine, mais ne répète pas à l'intérieur de sa structure les composants de dénomination communs. C'est ainsi que seul le RDN est utilisé pour identifier les objets gérés subordonnés.

Si la demande le spécifie, la réponse à l'action doit aussi signaler la classe d'objets gérés à laquelle appartient chaque objet géré rencontré dans le champ de la recherche.

La réponse à l'action peut se trouver distribuée dans de multiples réponses corrélées. Le gestionnaire peut reconstruire l'ensemble complet des objets gérés qui appartiennent au domaine d'application spécifié par combinaison du contenu de toutes les réponses corrélées.

L'emploi d'un nom distinctif relatif figé pour dénommer l'objet géré exploration permet d'accéder de manière uniforme à cet objet. C'est pourquoi il est fourni une relation de dénomination pour pouvoir assurer le confinement d'un objet géré exploration à l'intérieur d'un objet géré système à l'aide d'un nom distinctif relatif figé.

7.2 Objets d'Annuaire de connaissance de gestion

La force de l'Annuaire réside dans la fourniture d'informations de nature relativement stable. C'est pourquoi, dans le contexte de la gestion de la connaissance de gestion, des objets d'Annuaire servent à conserver la connaissance de répertoire relative aux systèmes gérés, comme par exemple, l'adresse de présentation d'une entité d'application de gestion-systèmes et la connaissance des définitions (voir Figure 1).

7.2.1 Objets d'Annuaire répertoires

L'exigence relative à l'identification des utilisateurs du service MIS peut être satisfaite par l'emploi d'objets d'Annuaire répertoires (*repertoire Directory objects*). Comme le montre la Figure 1, l'emploi de l'Annuaire permet de répondre à ce besoin pour les éléments suivants de la connaissance de répertoire:

- nom d'Annuaire d'une entité d'application de gestion-systèmes;
- adresse de présentation d'une entité d'application de gestion-systèmes;
- contextes d'application mis en œuvre par une entité d'application de gestion-systèmes;
- unités fonctionnelles mises en œuvre par une entité d'application de gestion-systèmes;
- profils de CMIP mis en œuvre par une entité d'application de gestion-systèmes;
- liste des objets gérés rendus visibles par une entité d'application de gestion-systèmes;
- étant donné un nom global d'objet géré, quelles entités d'application de gestion-systèmes peuvent être utilisées afin d'établir une association avec un système géré qui rend visible l'objet géré.

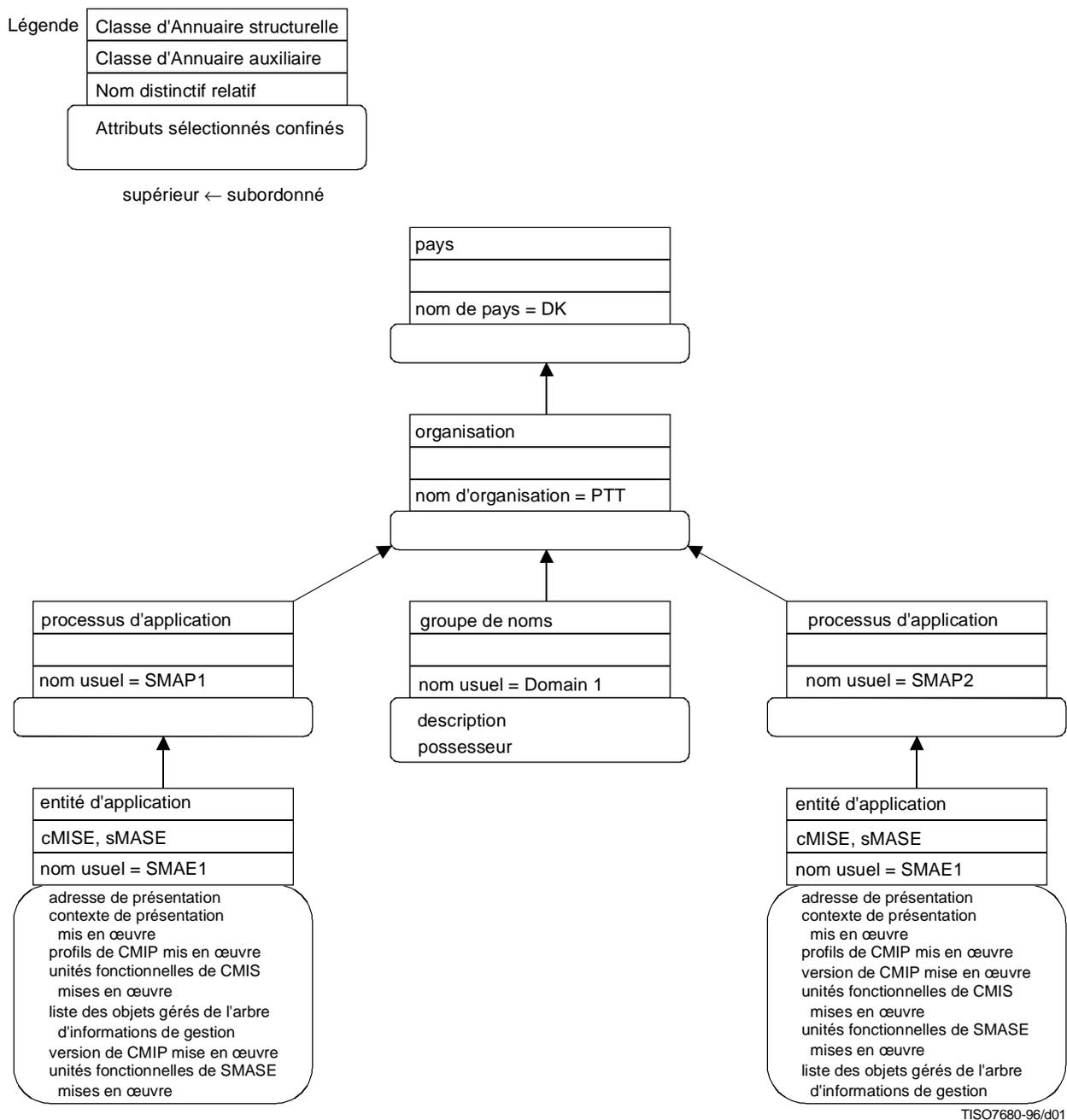


Figure 1 – Emploi de l'Annuaire pour la connaissance de répertoire

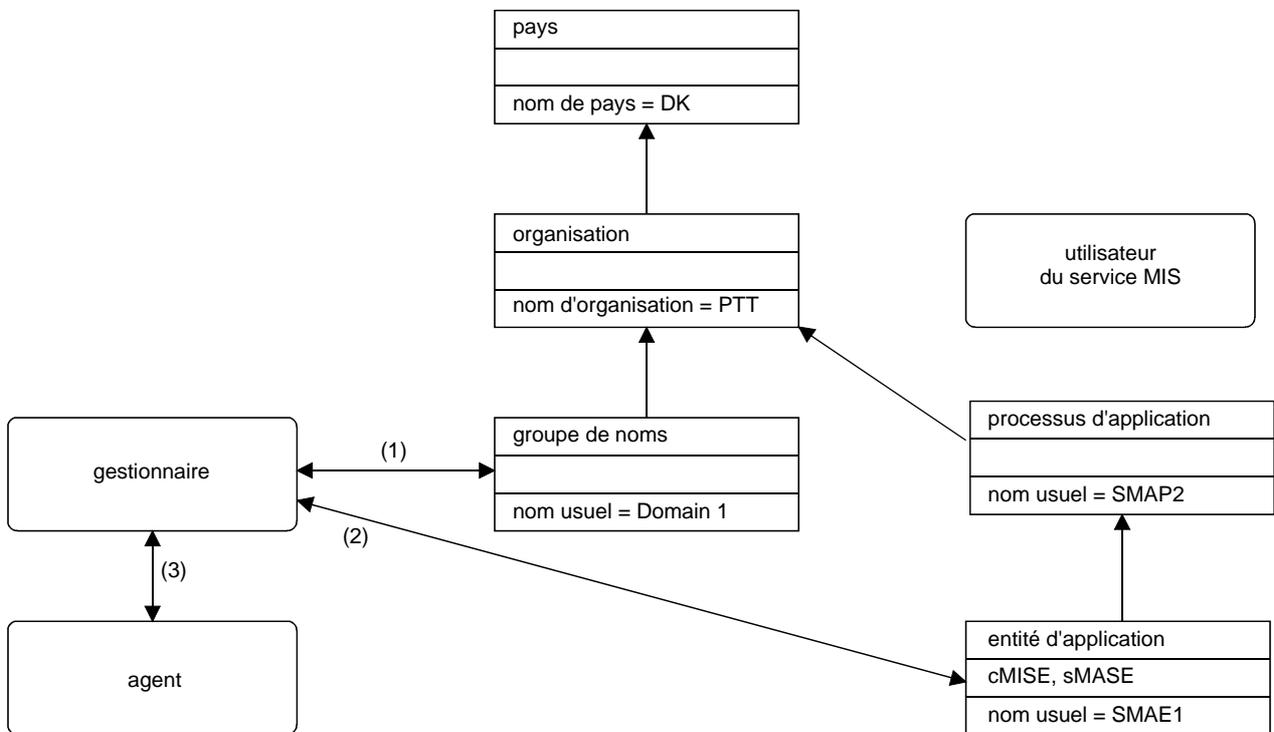
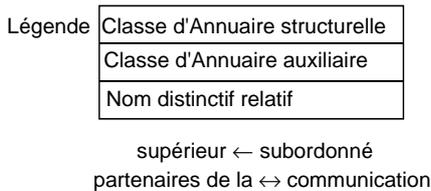
La connaissance de répertoire sur les entités d'application de gestion-systèmes des systèmes gestionnaires et gérés est donnée dans l'Annuaire par des instances de la classe d'objets d'Annuaire structurelle entités d'application. Les caractéristiques particulières des entités d'application de gestion-systèmes sont traitées par la définition des classes d'objets d'Annuaire auxiliaires éléments de service d'application de gestion-systèmes et éléments de service commun d'informations de gestion incluses dans une instance de la classe entité d'application. Cette instance peut facultativement contenir des attributs rendant disponibles l'adresse de présentation et les contextes d'application mis en œuvre.

La classe d'objets auxiliaires éléments de service d'application de gestion-systèmes donne les informations sur les unités fonctionnelles du service d'application de gestion-systèmes mises en œuvre. Elle peut aussi comporter un attribut pour identifier les classes d'objets gérés prises en compte. Cet attribut peut facultativement donner la liste des objets gérés d'une classe dont la durée de vie est relativement longue.

La classe d'objets auxiliaires éléments de service commun d'informations de gestion donne les informations sur les unités fonctionnelles du service commun d'informations de gestion, la version du protocole CMIP et les profils du CMIP mis en œuvre.

Un gestionnaire peut se servir des informations enregistrées dans l'Annuaire de la manière suivante (voir Figure 2):

- 1) Si un gestionnaire ignore de quels systèmes gérés il est en droit de prendre le contrôle, il lit la liste des entrées d'Annuaire qui représentent des systèmes gérés à partir de l'attribut membre d'une entrée d'Annuaire de la classe groupe de noms {countryName = DK, organizationName = PTT, commonName = Domain1 }.
- 2) Afin de découvrir les entités d'application de gestion-systèmes qui peuvent être utilisées pour établir une association avec le système géré sur lequel il désire opérer, le gestionnaire lance une opération de recherche dans l'Annuaire en utilisant le nom renvoyé dans la première phase. Il demande à recevoir l'adresse de présentation ainsi que, s'il le désire, d'autres informations relatives au protocole, telles que les contextes d'application ou les unités fonctionnelles de CMIS mis en œuvre. Afin de découvrir l'entité d'application de gestion-systèmes la plus adaptée aux besoins du gestionnaire, la recherche peut être fondée sur les valeurs d'attributs définis par la classe d'objets d'Annuaire structurelle entités d'application, le contexte d'application par exemple. Elle peut aussi être fondée sur les valeurs d'attributs définis par les classes d'objets d'Annuaire auxiliaires éléments de service d'application de gestion-systèmes et éléments de service commun d'informations de gestion, les unités fonctionnelles de SMASE ou la version de CMIP mises en œuvre par exemple.
- 3) Le gestionnaire choisit dans la liste d'entités d'application qui lui est renvoyée celle qui satisfait le mieux à ses besoins. Il se sert de l'adresse de présentation pour établir une association de présentation avec le système géré sur lequel il veut agir.



TISO7690-96/d02

Figure 2 – Exemple d'utilisation par un gestionnaire des objets d'Annuaire de répertoire

Un agent peut utiliser les informations enregistrées dans l'Annuaire de la manière suivante (voir Figure 3):

- 1) S'il doit expédier un rapport d'événement à un gestionnaire, l'agent obtient le titre de l'entité d'application du gestionnaire à partir de l'attribut destination du discriminateur d'émission d'événement. Il lit l'adresse de présentation de l'entité d'application de gestion-systèmes dans l'entrée d'Annuaire dont le nom est le titre de l'entité d'application du gestionnaire. (Dans l'exemple, le titre d'entité d'application du gestionnaire est {countryName = DK, organizationName = PTT, commonName = SMAP2, commonName = SMAE1}). Si l'agent s'intéresse aussi à d'autres attributs, comme les unités fonctionnelles de CMIS et de gestion-systèmes mises en œuvre, il peut aussi les lire dans l'entrée d'Annuaire.
- 2) L'agent établit une association de gestion avec l'entité de gestion-systèmes qui se trouve à l'adresse de présentation qu'il a obtenue.

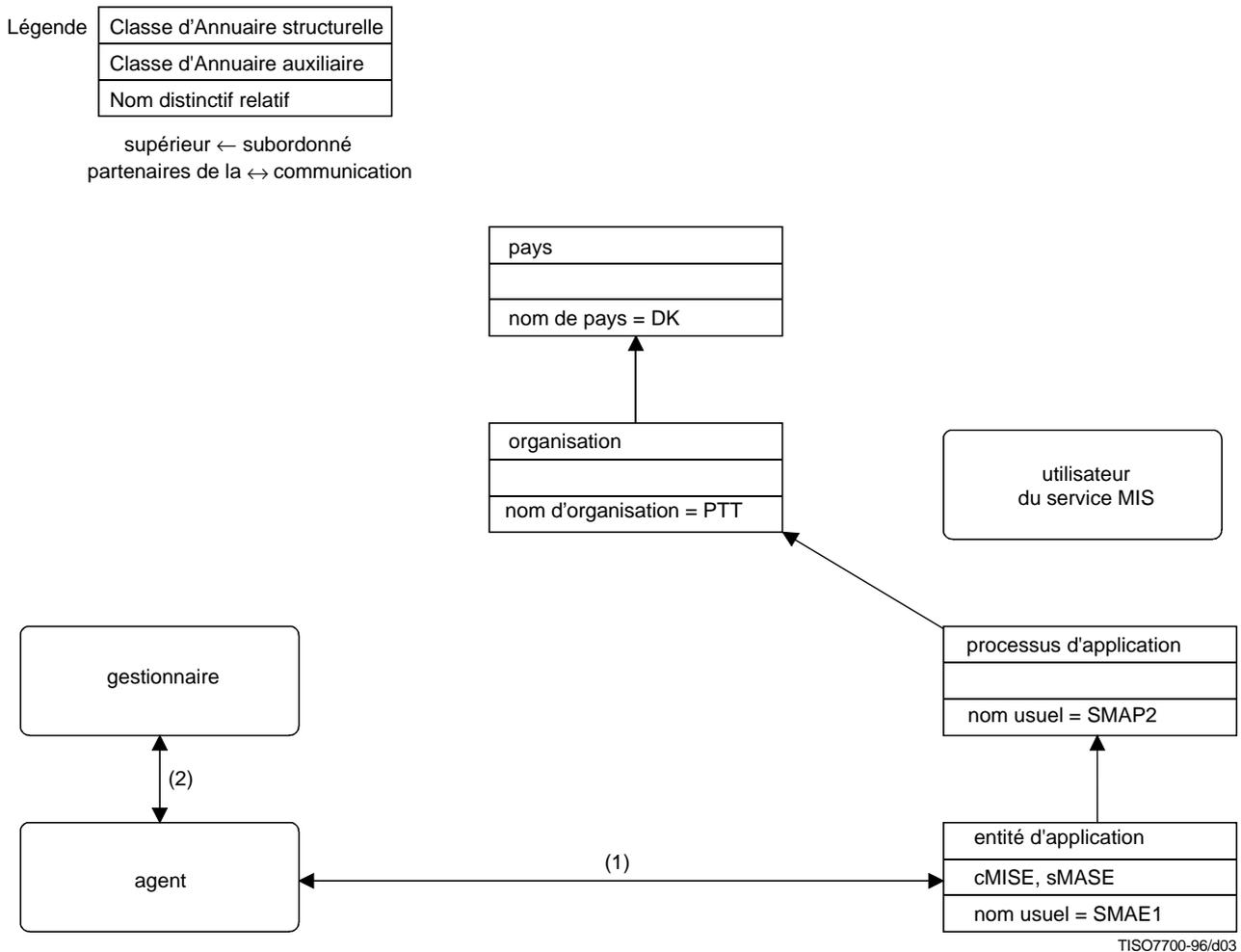


Figure 3 – Exemple d'utilisation par un agent des objets d'Annuaire de répertoire

Le nom distinctif de l'entrée d'Annuaire qui représente un système géré, par exemple une entrée d'Annuaire de la classe processus d'application, peut être utilisé pour construire les noms globaux des objets gérés rendus visibles par le système géré en question.

NOTE 1 – Le nom distinctif de l'entrée d'Annuaire qui représente un système géré peut être obtenu comme résultat final de la résolution de noms d'Annuaire pseudonymes.

Le choix de nom distinctif relatif employé pour nommer l'objet géré système a une influence sur les noms globaux de l'objet géré système lui-même et sur ceux de tous les objets gérés dont il définit directement ou indirectement les limites de confinement. Ces noms globaux peuvent se construire de la manière indiquée ci-dessous.

Si pour dénommer l'objet géré système on utilise l'attribut identificateur de système, le nom global d'un objet géré donné s'obtient en préfixant son nom distinctif local par le nom distinctif de l'entrée d'Annuaire qui représente le système géré et par le RDN de l'objet géré système. Si par exemple, comme le montre la Figure 4 a), le système géré est représenté par l'entrée d'Annuaire {countryName = DK, organizationName = PTT, commonName = SMAP1}, le nom global de l'objet géré discriminateur est {countryName = DK, organizationName = PTT, commonName = SMAP1, systemId = xyz, discriminatorId = 1}.

NOTE 2 – Cette approche est aussi utilisable si la racine de l'arbre local de confinement, au lieu d'être une instance de la classe d'objets gérés système, est une instance de la classe d'objets gérés réseau, comme le définit la Recommandation M.3100.

Si pour dénommer l'objet géré système on utilise l'attribut titre de système avec choix de la syntaxe nom distinctif, cet attribut prend la valeur de l'entrée d'Annuaire qui représente le système géré. Si par exemple, comme le montre la Figure 4 b), le système géré est représenté par l'entrée d'Annuaire {countryName = DK, organizationName = PTT, commonName = SMAP1}, le nom global de l'objet géré discriminateur est {systemTitle = {countryName = DK, organizationName = PTT, commonName = SMAP1}, discriminatorId = 1}.

Si pour dénommer l'objet géré système on utilise l'attribut titre de système avec choix de la syntaxe identificateur d'objet, cet identificateur doit être appliqué sur le nom distinctif de l'entrée d'Annuaire qui représente le système géré.

NOTE 3 – Le mécanisme qui permet d'appliquer l'identificateur d'objet sur un nom distinctif est hors du champ d'application de la présente Recommandation | Norme internationale.

Pour déterminer quelles entités d'application de gestion-systèmes peuvent servir à établir une association avec un système géré qui rend visible un objet géré donné, un gestionnaire peut procéder de la manière suivante (voir la Figure 4):

- 1) a) Si l'attribut identificateur de système est employé, le gestionnaire lance une opération de lecture de l'Annuaire en utilisant le nom global de l'objet géré. L'Annuaire renverra une erreur de nom (*nameError*) avec l'indication de problème pas d'objet de ce genre (*noSuchObject*). Cette indication d'erreur inclut le paramètre en concordance (*matched*) qui contient l'entrée de plus bas niveau de l'arbre DIT avec laquelle s'est établie la concordance. Cette concordance partielle de nom identifie l'entrée d'Annuaire qui représente le système géré.
- 1) b) Si l'attribut titre de système est employé, le gestionnaire extrait la valeur de cet attribut, identifiant l'entrée d'Annuaire qui représente le système géré.
- 2) Afin de trouver les entités d'application de gestion-systèmes utilisables pour établir une association avec le système géré, le gestionnaire lance alors une opération de recherche dans l'Annuaire en utilisant le nom de l'entrée d'Annuaire qui représente le système géré obtenu à l'étape 1) a) ou 1) b) selon le cas. Il demande à recevoir l'attribut adresse de présentation et, facultativement, d'autres attributs d'intérêt pour le protocole, comme par exemple les unités fonctionnelles de CMIS mises en œuvre. Afin de découvrir celle des entités d'application de gestion-systèmes qui est la plus adaptée aux besoins du gestionnaire, la recherche peut se fonder sur les valeurs d'attributs définis par la classe d'objets d'Annuaire structurelle entités d'application, contextes d'application mis en œuvre par exemple. Elle peut aussi se fonder sur les valeurs d'attributs définis par les classes d'objets d'Annuaire auxiliaires éléments de service d'application de gestion-systèmes et éléments commun d'informations de gestion, unités fonctionnelles de SMASE, de CMIS ou version de CMIP mises en œuvre par exemple.
- 3) Le gestionnaire choisit une des entités d'application de gestion-systèmes dans la liste des entités convenables qui résulte de l'opération de recherche. Il se sert de son adresse de présentation et des autres informations intéressant le protocole qu'il a éventuellement reçues pour établir une association de gestion.

7.2.2 Objets d'Annuaire définitions

Le présent modèle explique comment conserver les définitions d'informations de gestion dans un arbre d'Annuaire de connaissance des définitions (*definition knowledge directory tree*), fondé sur le concept d'arbre d'identificateurs d'objets présenté dans la Rec. UIT-T X.680 | ISO/CEI 8824-1. Le modèle permet en outre d'insérer dans l'arbre d'Annuaire de connaissance des définitions d'informations auxquelles n'ont pas été attribués d'identificateurs d'objet.

Les entrées qui dans l'Annuaire représentent l'arbre d'Annuaire de connaissance des définitions appartiennent à la classe d'Annuaire structurelle informations enregistrées. De plus, en fonction du type d'information à enregistrer, il est possible de définir des classes d'Annuaire auxiliaires contenant d'autres informations. Des instances de ces classes auxiliaires peuvent trouver place dans les entrées de la classe informations enregistrées.

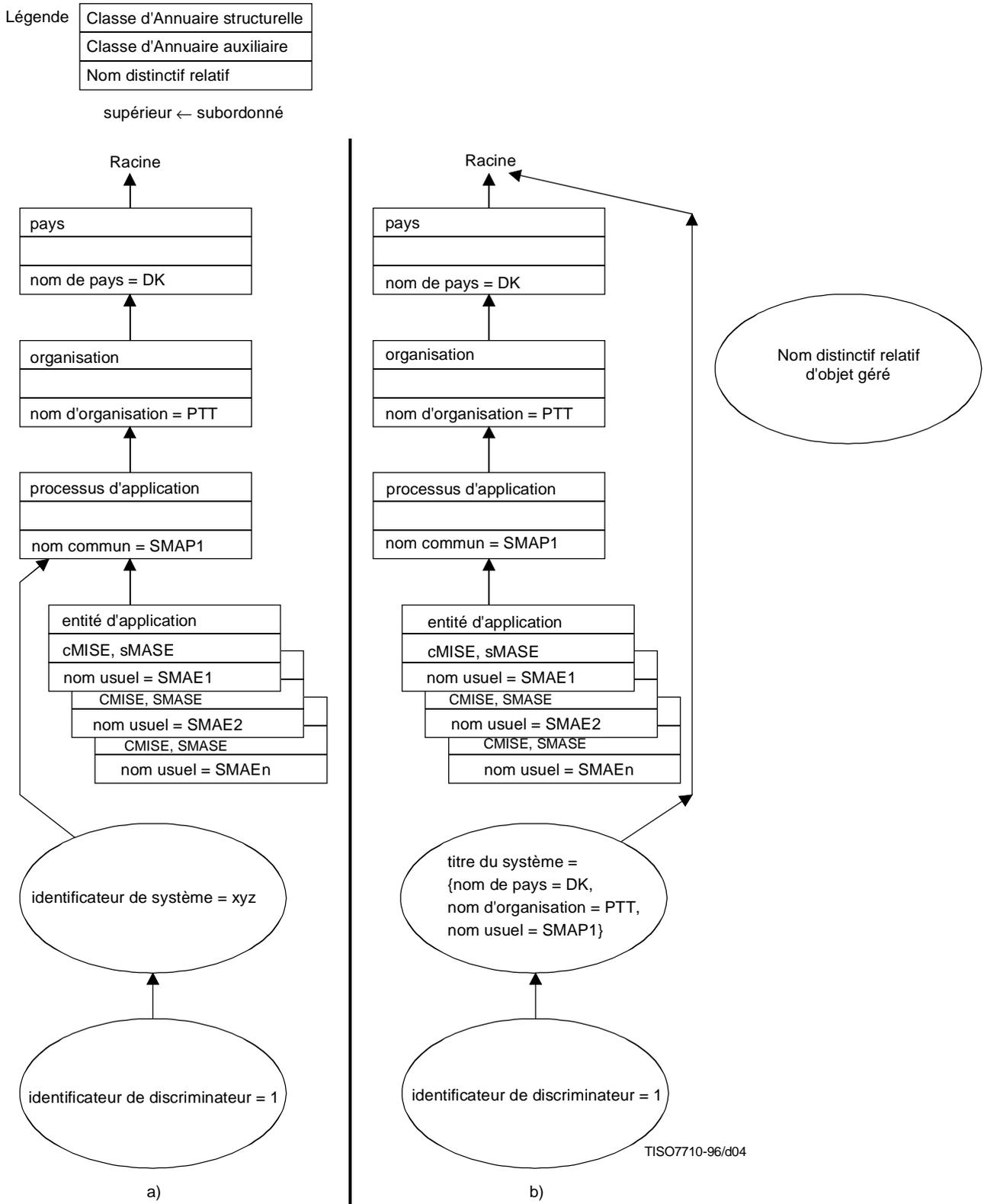
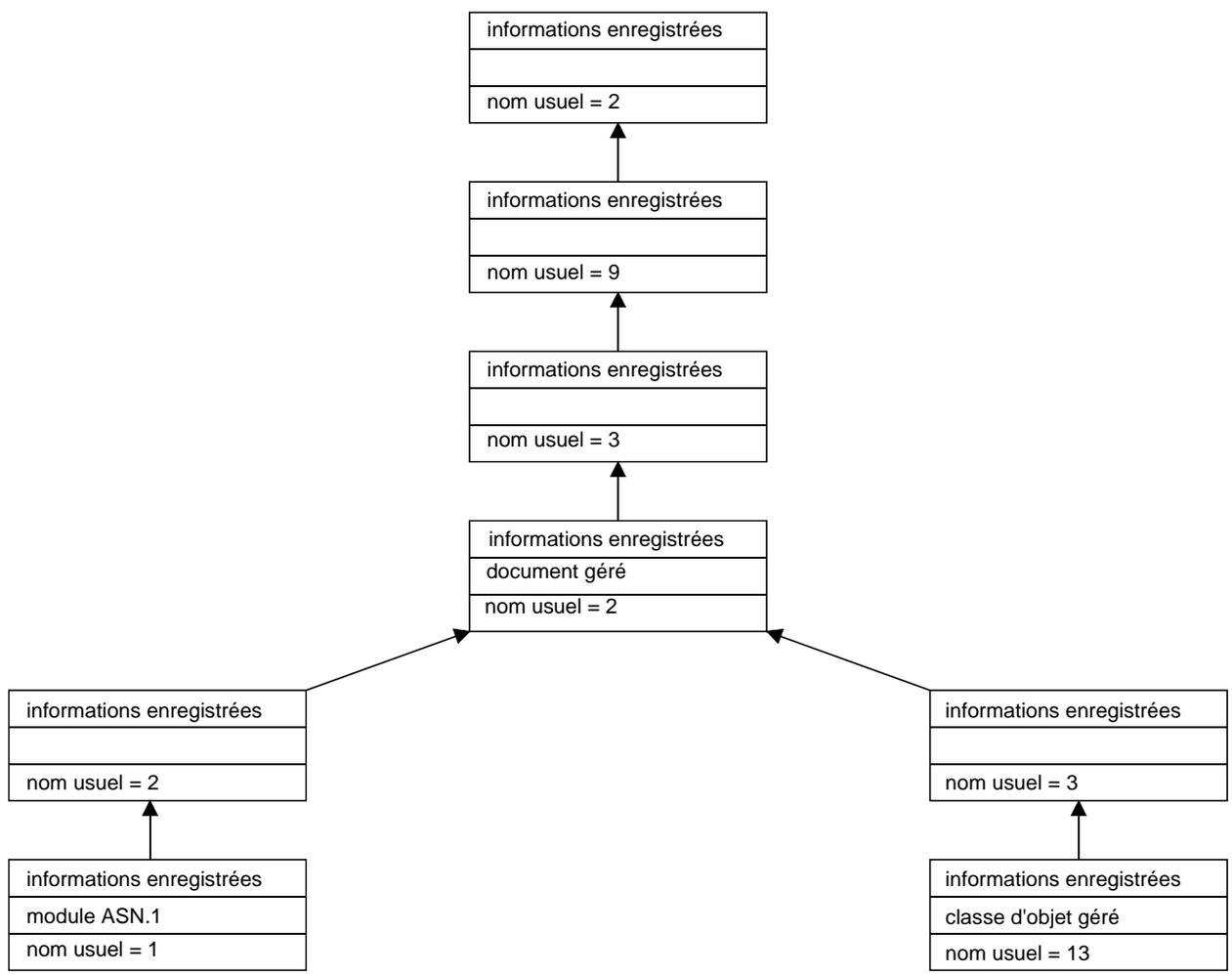
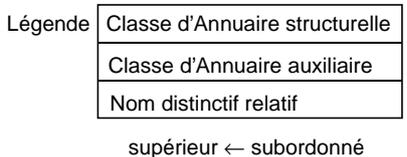


Figure 4 – Construction d'un nom global MKM

La construction du nom distinctif d'une entrée informations enregistrées à partir d'un identificateur d'objet donné se fait directement. La Figure 5 présente les noms distinctifs pour deux des définitions de la Rec. X.721 du CCITT | ISO/CEI 10165-2:

- la classe d'objets gérés systèmes, enregistrée sous {joint-iso-ccitt(2) ms(9) smi(3) part2(2) managedObjectClass(3) 13};
- l'attribut Module ASN.1, enregistré sous {2 9 3 2 2 1}.



TISO7720-96/d05

Figure 5 – Exemple d'arbre d'Annuaire de connaissance des définitions

La valeur d'identificateur d'objet {2 9 3 2 3 13} est appliquée sur le nom distinctif {commonName = 2, commonName = 9, commonName = 3, commonName = 2, commonName = 3, commonName = 13}.

NOTE – Un préfixe dépendant de l'emplacement où s'effectue l'inclusion de l'arbre d'Annuaire de connaissance des définitions dans l'arbre d'informations d'Annuaire global doit être ajouté à ce nom distinctif (voir l'Annexe H).

7.3 Relations entre les exigences relatives à la connaissance de gestion et le modèle

Le Tableau 2 montre la relation qui existe entre les exigences relatives à la connaissance de gestion et les objets de connaissance de gestion définis dans le modèle. Une croix indique que l'exigence associée est satisfaite par les objets de connaissance de gestion correspondants.

Tableau 2 – Exigences et modèle

Exigence de connaissance	Objets de connaissance de gestion				
	Objets gérés répertoires	Objets gérés définitions	Objets gérés explorations	Objets d'Annuaire répertoires	Objets d'Annuaire définitions
Connaissance des classes d'objets gérés	X	–	–	X (Note 1)	–
Connaissance des instances de classes d'objets gérés	–	–	X (Note 2)	X (Note 3)	–
Connaissance des relations (Note 4)	X (Note 5)	–	–	–	–
Connaissance de l'utilisateur du service MIS	X (Note 6)	–	–	X	–
Connaissance des définitions des informations de gestion	–	X (Note 7)	–	–	X (Note 7)

NOTES

1 Pour l'instant, la présente Recommandation | Norme internationale ne définit d'objets d'Annuaire répertoires que pour représenter les classes d'objets gérés mises en œuvre par un système géré.

2 Seuls peuvent être déterminés les objets gérés rendus visibles par un système géré donné.

3 Il n'est prévu d'enregistrer que les objets gérés susceptibles de persister relativement durablement.

4 Pour l'instant, il n'a pas été défini d'objets de connaissance de gestion spécifiques pour la représentation de la connaissance d'une instance de relation.

5 Pour l'instant, les seules informations représentées sont l'ensemble des classes de relations gérées et l'ensemble des relations de dénomination mises en œuvre par un système géré.

6 Pour l'instant, seule est fournie l'information relative aux profils de CMIP mis en œuvre.

7 Pour l'instant, la connaissance des définitions ne prend en charge que les modèles GDMO et les modules ASN.1.

8 Définitions génériques

8.1 Objets gérés

La présente Recommandation | Norme internationale définit des classes d'objets gérés pour la connaissance de gestion et les caractéristiques associées.

8.1.1 Objets gérés pour la connaissance de répertoire

8.1.1.1 Objet géré répertoire de classe d'objets gérés

Les objets gérés répertoires de classe d'objets gérés (*managed object class repertoire managed objects*) identifient pour une classe donnée d'objets gérés les capacités mises en œuvre.

La classe d'objets gérés répertoire de classe d'objets gérés est définie comme une sous-classe de la classe d'objets gérés sommet définie par la Rec. X.721 du CCITT | ISO/CEI 10165-2. Ses attributs obligatoires, utilisables en lecture seule, sont:

- identificateur de répertoire de classe d'objets gérés (*managed object class repertoire Id*);
- liste des blocs de propriétés conditionnels mis en œuvre (*supported conditional package list*).

Son attribut facultatif, utilisable en lecture seule, est:

- valeurs implémentées (*implemented values*).

L'objet géré répertoire de classe d'objets gérés prend en charge les notifications suivantes:

- création d'objet (*object creation*);
- suppression d'objet (*object deletion*);
- modification de valeur d'attribut (*attribute value change*).

8.1.1.2 Objet géré répertoire

Les objets gérés répertoires (*repertoire managed objects*) identifient les classes d'objets gérés, les classes de relations gérées, les relations de dénomination et les profils de CMIP mis en œuvre dans un système.

Un objet géré répertoire a la faculté de servir de supérieur pour des objets subordonnés, chacun d'entre eux transmettant la connaissance concernant la mise en œuvre par une implémentation d'une classe particulière d'objets gérés.

La classe d'objets gérés répertoires est définie comme une sous-classe de la classe d'objets gérés sommet définie par la Rec. X.721 du CCITT | ISO/CEI 10165-2. Ses attributs obligatoires, utilisables en lecture seule, sont:

- a) identificateur de répertoire (*repertoire Id*);
- b) liste des relations de dénomination mises en œuvre (*supported name binding list*);
- c) liste des classes d'objets gérés mises en œuvre (*supported managed object class list*);
- d) liste des classes de relations mises en œuvre (*supported relationship class list*).

Son attribut facultatif, utilisable en lecture seule, est:

- e) profils de CMIP mis en œuvre (*supported CMIP profiles*).

L'objet géré répertoire prend en charge les notifications suivantes:

- a) création d'objet;
- b) suppression d'objet;
- c) modification de valeur d'attribut.

8.1.2 Objets gérés pour la connaissance des définitions

8.1.2.1 Objet géré modèle d'action

L'objet géré modèle d'action (*action template managed object*) représente une spécification GDMO d'action.

La classe d'objets gérés modèles d'action est définie comme une sous-classe de la classe d'objets gérés modèles. Ses attributs obligatoires, utilisables en lecture seule, sont:

- a) comportement (*behaviour*);
- b) mode confirmé (*mode confirmed*);
- c) paramètres (*parameters*);
- d) avec syntaxe d'informations (*with information syntax*);
- e) avec syntaxe de réponse (*with reply syntax*);
- f) enregistré sous (*registered as*).

8.1.2.2 Objet géré module ASN.1

L'objet géré module ASN.1 (*ASN.1 module managed object*) représente le contenu d'un module ASN.1 sous forme textuelle.

La classe d'objets gérés modules ASN.1 est définie comme une sous-classe de la classe d'objets gérés sommet définie par la Rec. X.721 du CCITT | ISO/CEI 10165-2. Ses attributs obligatoires, utilisables en lecture seule, sont:

- a) contenu de module ASN.1 (*ASN.1 module contents*);
- b) version d'ASN.1 (*ASN.1 version*);
- c) référence de module (*module reference*);
- d) facultativement enregistré sous (*optionally registered as*).

L'objet géré module ASN.1 dispose aussi d'action pour demander le contenu d'un module ASN.1.

Pour l'envoi de la version textuelle du contenu du module ASN.1, la réponse à l'action peut utiliser des réponses corrélées si l'association le permet.

8.1.2.3 Objet géré modèle de groupe d'attributs

L'objet géré modèle de groupe d'attributs (*attribute group template managed object*) représente une spécification GDMO de groupe d'attributs.

La classe d'objets gérés modèles de groupe d'attributs est définie comme une sous-classe de la classe d'objets gérés modèles. Ses attributs obligatoires, utilisables en lecture seule, sont:

- a) éléments du groupe (*group elements*);
- b) figé (*fixed*);
- c) description (*description*);
- d) enregistré sous.

8.1.2.4 Objet géré modèle d'attribut

L'objet géré modèle d'attribut (*attribute template managed object*) représente une spécification GDMO d'attribut.

La classe d'objets gérés modèles d'attributs est définie comme une sous-classe de la classe d'objets gérés modèles. Ses attributs obligatoires, utilisables en lecture seule, sont:

- a) dérivé ou avec choix de syntaxe (*derived or with syntax choice*);
- b) concorde si (*matches for*);
- c) comportement;
- d) paramètres;
- e) facultativement enregistré sous.

8.1.2.5 Objet géré modèle de comportement

L'objet géré modèle de comportement (*behaviour template managed object*) représente une spécification GDMO de comportement.

La classe d'objets gérés modèles de comportement est définie comme une sous-classe de la classe d'objets gérés modèles. Son attribut obligatoire, utilisable en lecture seule, est:

- défini comme (*defined as*).

L'objet géré modèle de comportement dispose aussi d'une action pour demander la description de comportement.

Pour l'envoi de la version textuelle de la description de comportement, la réponse à l'action peut utiliser des réponses corrélées si l'association le permet.

8.1.2.6 Objet géré document

L'objet géré document (*document managed object*) représente un document contenant des spécifications GDMO et ASN.1. Il se comporte donc comme un réceptacle d'objets gérés modèles.

La classe d'objets gérés documents est définie comme une sous-classe de la classe d'objets gérés sommet définie par la Rec. X.721 du CCITT | ISO/CEI 10165-2. Ses attributs obligatoires, utilisables en lecture seule, sont:

- a) nom de document (*document name*);
- b) identificateur d'objet de document (*document object identifier*).

Le bloc de propriétés conditionnel spécification de document est présent si l'objet géré document prend en charge la représentation textuelle de la totalité des spécifications GDMO et ASN.1 contenues dans le document. S'il est présent, il comporte l'attribut suivant, utilisable en lecture seule:

- c) spécification (*specification*).

Le bloc de propriétés conditionnel dispose aussi d'une action pour demander la spécification. Si ce bloc est présent, l'objet géré document répond aux actions demandant à obtenir la représentation textuelle des spécifications GDMO et ASN.1 contenues dans le document.

Pour l'envoi de la version textuelle de la spécification, la réponse à l'action peut utiliser des réponses corrélées si l'association le permet.

Si ce bloc de propriétés n'est pas présent, il est possible de rechercher les versions textuelles des spécifications dans des objets gérés modèles subordonnés.

8.1.2.7 Objet géré modèle de classe d'objets gérés

L'objet géré modèle de classe d'objets gérés (*managed object class template*) représente une spécification GDMO de classe d'objets gérés.

La classe d'objets gérés modèles de classe d'objets gérés est définie comme une sous-classe de la classe d'objets gérés modèles. Ses attributs obligatoires, utilisables en lecture seule, sont:

- a) dérivé de (*derived from*);
- b) caractérisé par (*characterized by*);
- c) blocs de propriétés conditionnels;
- d) enregistré sous.

8.1.2.8 Objet géré modèle de relation de dénomination

L'objet géré modèle de relation de dénomination (*name binding template managed object*) représente une spécification GDMO de relation de dénomination.

La classe d'objets gérés modèles de relations de dénomination est définie comme une sous-classe de la classe d'objets gérés modèles. Ses attributs obligatoires, utilisables en lecture seule, sont:

- a) classe d'objets gérés subordonnée (*subordinate managed object class*);
- b) dénommé par classe d'objets supérieure (*named by superior object class*);
- c) avec l'attribut (*with attribute*);
- d) comportement;
- e) création (*create*);
- f) suppression (*delete*);
- g) enregistré sous.

8.1.2.9 Objet géré modèle de notification

L'objet géré modèle de notification (*notification template managed object*) représente une spécification GDMO de notification.

La classe d'objets gérés modèles de notifications est définie comme une sous-classe de la classe d'objets gérés modèles. Ses attributs obligatoires, utilisables en lecture seule, sont:

- a) comportement;
- b) mode confirmé;
- c) paramètres;
- d) avec la syntaxe d'informations;
- e) et les identificateurs d'attribut (*and attribute Id's*);
- f) avec la syntaxe de réponse;
- g) enregistré sous.

8.1.2.10 Objet géré modèle de bloc de propriétés

L'objet géré modèle de bloc de propriétés (*package template managed object*) représente une spécification GDMO de bloc de propriétés.

La classe d'objets gérés modèles de blocs de propriétés est définie comme une sous-classe de la classe d'objets gérés modèles. Ses attributs obligatoires, utilisables en lecture seule, sont:

- a) comportement;
- b) attributs;
- c) groupes d'attributs;
- d) actions (*actions*);
- e) notifications;
- f) facultativement enregistré sous.

8.1.2.11 Objet géré modèle de paramètre

L'objet géré modèle de paramètre (*parameter template managed object*) représente une spécification GDMO de paramètre.

La classe d'objets gérés modèles de paramètres est définie comme une sous-classe de la classe d'objets gérés modèles. Ses attributs obligatoires, utilisables en lecture seule, sont:

- a) contexte (*context*);
- b) syntaxe ou attribut (*syntax or attribute*);
- c) comportement;
- d) facultativement enregistré sous.

8.1.2.12 Objets gérés modèles

La classe d'objets gérés modèles (*template managed objects*) est celle dont peuvent être dérivées les classes d'objets servant à représenter les spécifications GDMO. Si seules sont mises en œuvre des représentations textuelles des modèles GDMO, alors sont créés des objets gérés appartenant à la classe d'objets gérés modèles. Cette classe fournit l'attribut de dénomination pour les objets gérés qui représentent des modèles. Par le moyen de la spécification d'une seule relation de dénomination, elle leur permet d'appartenir au domaine de confinement des objets gérés documents.

La classe d'objets gérés modèles est une sous-classe de la classe d'objets gérés sommet définie par la Rec. X.721 du CCITT | ISO/CEI 10165-2. Son attribut obligatoire, utilisable en lecture seule, est:

- a) nom de modèle (*template name*).

L'objet de gestion modèle peut aussi disposer du bloc de propriétés optionnel définition de modèle. S'il est présent, celui-ci possède l'attribut suivant, utilisable en lecture seule:

- b) définition de modèle (*template definition*).

Le bloc de propriétés conditionnel dispose aussi d'une action pour demander la définition de modèle. Si ce bloc est présent, l'objet géré modèle répond aux actions demandant à obtenir la représentation textuelle de la définition de modèle.

Pour l'envoi de la version textuelle de la spécification de modèle, la réponse à l'action peut utiliser des réponses corrélées si l'association le permet.

8.1.3 Objets gérés pour la connaissance des instances

8.1.3.1 Objet géré exploration

L'objet géré exploration (*discovery managed object*) sert à déterminer l'ensemble, ou une partie sélectionnée, des sous-arbres d'objets gérés appartenant à l'arbre de dénomination d'un système géré.

La classe d'objets gérés explorations est une sous-classe de la classe d'objets gérés sommet définie par la Rec. X.721 du CCITT | ISO/CEI 10165-2. Son attribut obligatoire, utilisable en lecture seule, est:

- identificateur d'exploration (*discovery Id*).

L'emploi d'une relation de dénomination entre les objets exploration et système avec une valeur figée du nom distinctif relatif offre un mode général pour l'accès à l'objet géré exploration.

Cet objet répond aux actions comportant des demandes relatives aux sous-arbres d'objets de gestion appartenant à l'arbre de dénomination d'un système géré. La demande spécifie un objet de base et le domaine de recherche exigé.

La réponse est constituée d'un ensemble de structures arborescentes qui, toutes ensemble, couvrent le domaine demandé en dessous de l'objet de base demandé. Chaque arbre porte le nom distinctif de sa racine, sans pourtant répéter les composantes communes de dénomination de sa structure. C'est ainsi que seul le RDN est utilisé pour identifier les objets gérés subordonnés.

8.2 Relations de dénomination

8.2.1 Relations de dénomination pour la connaissance du répertoire

8.2.1.1 Répertoire de classe d'objets gérés – Répertoire

Cette relation de dénomination permet le confinement des objets gérés répertoires de classe d'objets gérés, ainsi que de leurs spécialisations, dans des objets gérés répertoires ou dans leurs spécialisations.

ISO/CEI 10164-16 : 1997 (F)

La dénomination doit employer l'attribut identificateur de répertoire de classe d'objets gérés, qui prend pour valeur l'identificateur d'objet de la classe d'objets gérés auquel le répertoire est associé.

8.2.1.2 Répertoire – Système

Cette relation de dénomination permet le confinement des objets gérés répertoires, ainsi que de leurs spécialisations, dans des objets gérés systèmes ou dans leurs spécialisations. La dénomination doit employer l'attribut identificateur de répertoire.

L'emploi d'un nom distinctif relatif figé donne une méthode générale d'accès à un objet géré répertoire. La présente relation de dénomination en donne le moyen si la valeur affectée à l'attribut identificateur de répertoire est «REPI».

8.2.2 Relations de dénomination pour la connaissance des définitions

8.2.2.1 Module ASN.1 – Document

Cette relation de dénomination permet le confinement des objets gérés modules ASN.1, ainsi que de leurs spécialisations, dans des objets gérés documents ou dans leurs spécialisations. L'attribut référence de module fournit l'identification.

8.2.2.2 Document – Système

Cette relation de dénomination permet le confinement des objets documents, ainsi que de leurs spécialisations, dans des objets gérés systèmes ou dans leurs spécialisations. L'attribut identificateur d'objet de document fournit l'identification.

8.2.2.3 Document – Système2

Cette relation de dénomination permet le confinement des objets documents, ainsi que de leurs spécialisations, dans des objets gérés systèmes ou dans leurs spécialisations. L'attribut nom de document fournit l'identification.

8.2.2.4 Modèle – Document

Cette relation de dénomination permet le confinement des objets modèles, ainsi que de leurs spécialisations, dans des objets gérés documents ou dans leurs spécialisations. L'attribut nom de modèle fournit l'identification.

8.2.3 Relations de dénomination pour la connaissance des instances

8.2.3.1 Exploration – Système

Cette relation de dénomination permet le confinement des objets gérés explorations, ainsi que de leurs spécialisations, dans des objets gérés système ou dans leurs spécialisations.

La dénomination doit employer l'attribut identificateur d'exploration. L'emploi d'un nom distinctif relatif figé donne une méthode générale d'accès à un objet géré d'exploration. La présente relation de dénomination en donne le moyen si la valeur affectée à l'attribut identificateur d'exploration est "DSC1".

Dans la relation de subordination à un objet géré système donné, l'attribut identificateur d'exploration ne peut prendre qu'une seule valeur, "DSC1".

8.3 Actions

La présente Recommandation | Norme internationale définit un seul type d'action pour la connaissance des définitions:

- obtention de représentation textuelle (*get textual representation*);

et un seul type d'action pour la connaissance des instances:

- recherche d'arbre d'informations de gestion (*management information tree search*).

Pour toute classe d'objets gérés qui inclut l'un ou l'autre de ces types d'action, il appartient aux responsables de la définition de la classe d'objets gérés de détailler complètement les exigences qui portent sur un objet géré de cette classe concernant la réponse à ces actions.

8.3.1 Actions pour la connaissance des définitions

L'argument de l'action d'obtention de représentation textuelle contient le paramètre identificateur d'attribut. La réponse est la représentation textuelle de la valeur de l'attribut identifié. Elle est envoyée sous forme d'une séquence d'images de lignes.

La réponse à l'action peut se trouver répartie dans de multiples réponses corrélées.

Les paramètres d'action sont présentés dans les paragraphes suivants.

8.3.1.1 Identificateur d'attribut

Le paramètre identificateur d'attribut (*attribute Id*) spécifie l'attribut de l'objet géré dont la valeur doit être renvoyée. La définition de la classe d'objets de gestion précise quels sont les attributs qu'il est permis de spécifier.

8.3.1.2 Représentation textuelle

Le paramètre représentation textuelle (*textual representation*) contient la réponse ou, lorsque des réponses corrélées sont utilisées, une portion continue de la réponse.

8.3.2 Actions pour la connaissance des instances

L'argument de l'action de recherche d'arbre d'informations de gestion contient comme paramètres l'objet de base, le domaine de la recherche ainsi qu'un paramètre facultatif de demande de classe. La réponse se compose d'un ensemble de structures arborescentes qui, toutes ensemble, couvrent le domaine demandé en dessous de l'objet de base demandé. Chaque arbre porte le nom distinctif de sa racine, sans toutefois répéter les composants communs de noms dans la structure. Ainsi, seul le RDN sert à identifier les objets gérés subordonnés.

Si la demande le spécifie, la réponse à l'action donnera aussi la classe d'objets gérés à laquelle appartient chaque objet géré présent dans le domaine de recherche.

La réponse peut se trouver répartie dans de multiples réponses corrélées.

Les paramètres d'action sont présentés dans les paragraphes suivants.

8.3.2.1 Base

Le paramètre base (*base*) spécifie le point de départ du sous-arbre à explorer dans le système géré.

8.3.2.2 Domaine d'exploration

Le paramètre domaine d'exploration (*discovery scope*) spécifie l'étendue du sous-arbre à explorer dans le système géré. Sa syntaxe est la même que celle de domaine dans le CMIP.

8.3.2.3 Demande de classe

Le paramètre demande de classe (*class request*) est une variable booléenne. Si sa valeur est «vrai», la réponse à l'action doit inclure les informations relatives aux classes d'objets gérés. Sa valeur par défaut est «faux».

8.3.2.4 Arbre de dénomination

Le paramètre arbre de dénomination (*rooted name tree*) contient l'identification de l'objet géré racine. Il peut aussi contenir une structure arborescente définie récursivement pour décrire un sous-arbre requis dans la demande de recherche d'arbre d'informations de gestion. La structure arborescente définie récursivement contient le RDN de chaque objet géré découvert. Si la demande en a été faite, la classe de chaque objet géré découvert doit aussi être fournie.

8.4 Compatibilité

Il est permis aux définitions de classes d'objets gérés d'importer selon leurs besoins des spécifications d'objets gérés, d'actions, de paramètres ou de types d'attributs définies dans la présente Recommandation | Norme internationale. L'opération s'effectue par référence aux modèles qui y sont définis, selon le mécanisme décrit dans la Rec. X.722 du CCITT | ISO/CEI 10165-4.

8.5 Définitions génériques provenant de la fonction de gestion d'objets

La présente Recommandation | Norme internationale fait usage des définitions génériques suivantes qui proviennent de la Rec. X.730 du CCITT | ISO/CEI 10164-1:

- notification de modification de valeur d'attribut (*attribute value change notification*);
- notification de création d'objet (*object creation notification*);
- notification de suppression d'objet (*object deletion notification*).

8.6 Objets d'Annuaire

8.6.1 Objets d'Annuaire pour les répertoires

Les définitions des classes d'objets d'Annuaire auxiliaires éléments de service d'application de gestion-systèmes (*systems management application service element*) et éléments de service commun d'informations de gestion (*common management information service element*) les placent dans les entrées d'Annuaire qui représentent des entités d'application faisant usage de la gestion-systèmes OSI. Les paragraphes suivants précisent comment elles doivent être utilisées, en même temps que la classe d'objets d'Annuaire structurelle entités d'application, dans une règle de contenu de l'arbre DIT (voir 6.12, de la Rec. UIT-T X.521 | ISO/CEI 9594-7).

La classe d'objets d'Annuaire auxiliaire éléments de service d'application de gestion-systèmes comporte obligatoirement l'attribut "unités fonctionnelles de SMASE mises en œuvre" (*supported SMASE functional units*). Il spécifie les unités fonctionnelles qu'une entité d'application de gestion-systèmes met en œuvre et les rôles auxquels elles participent. Cette classe comporte facultativement les attributs "met en œuvre les noms globaux MKM" (*supports MKM global names*) et "liste des objets gérés de l'arbre d'informations de gestion" (*management information tree managed object list*). Le premier attribut indique si une entité d'application de gestion-systèmes est en mesure de traiter les noms globaux d'objets gérés en accord avec les prescriptions de la présente Recommandation | Norme internationale. Le second attribut précise la liste des noms des objets gérés rendus visibles par l'entité d'application de gestion-systèmes en question. Dans le cas d'objets à courte durée de vie, le seul enregistrement prévu dans la liste est celui de la classe d'objets gérés.

La classe d'objets d'Annuaire auxiliaire éléments de service commun d'informations de gestion comporte obligatoirement les attributs "version de CMIP mise en œuvre" (*supported CMIP version*), "profils de CMIP mis en œuvre" (*supported CMIP profiles*) et "unités fonctionnelles de CMIS mises en œuvre" (*supported CMIS functional units*). Le premier attribut indique quelles versions du protocole commun d'informations de gestion l'entité d'application de gestion-systèmes est en mesure de traiter. Les second et troisième attributs donnent respectivement les profils normalisés de CMIP et les unités fonctionnelles de CMIS qu'une entité d'application de gestion-systèmes met en œuvre.

8.6.2 Objets d'Annuaire pour les définitions

La classe d'Annuaire structurelle informations enregistrées (*registered information*) représente un objet d'information auquel a été attribué un identificateur d'objet en accord avec la Rec. UIT-T X.680 | ISO/CEI 8824-1. Elle comporte obligatoirement l'attribut "nom usuel" (*commonName*) qui sert à dénommer les instances de cette classe. Il a pour valeur le composant final de l'identificateur d'objet qui a été attribué à l'objet géré représenté.

Les instances de cette classe comportent facultativement les attributs "forme de nom" (*name form*), "statut des informations" (*information status*) et "informations complémentaires" (*additional information*). L'attribut "forme de nom" spécifie le nom symbolique de la valeur en forme numérique du composant de l'identificateur d'objet conservé dans l'attribut nom usuel. L'attribut "statut des informations" spécifie si l'information est:

- active, c'est-à-dire que la définition de l'information, figée, est utilisable;
- supprimée, c'est-à-dire que la définition de l'information, périmée, ne devrait pas faire l'objet de références par de nouvelles définitions d'information;
- préliminaire, c'est-à-dire que la définition de l'information n'est donnée qu'à titre d'information, des changements pouvant intervenir dans l'avenir (par exemple du fait de la progression d'un document normatif du niveau projet de comité, *committee draft*, au niveau projet de norme internationale, *draft international standard*).

Des informations complémentaires relatives aux informations de définitions, telles que date de création et commentaire sur la définition peuvent facultativement être conservées dans l'attribut "informations complémentaires".

NOTE – Il est possible que soient définies des classes d'Annuaire auxiliaires additionnelles à inclure dans les entrées d'Annuaire informations enregistrées.

La définition de la classe d'Annuaire auxiliaire documents (*document*) la place dans les entrées d'Annuaire de la classe d'Annuaire structurelle informations enregistrées. Cette classe représente un document qui contient des spécifications GDMO. Elle comporte obligatoirement les attributs "nom de document" (*document name*), "identificateur d'objet de document" (*document object identifier*) et "spécification" (*specification*). L'attribut "nom de document" donne le nom du document représenté par une instance, par exemple Rec. X.721 du CCITT | ISO/CEI 10165-2. La valeur de l'attribut "identificateur d'objet de document" est celle de l'identificateur d'objet attribué au document, en accord avec les prescriptions de la Rec. X.722 du CCITT | ISO/CEI 10165-4. L'attribut "spécification" peut facultativement contenir une représentation des spécifications GDMO présentes dans le document.

C'est de la classe d'Annuaire auxiliaire modèles (*template*) qu'il est permis de dériver les classes d'Annuaire auxiliaires qui représentent des modèles GDMO. Elle comporte obligatoirement l'attribut "nom de modèle" (*template name*) dont la valeur est celle de l'étiquette du modèle GDMO qui est représenté. Elle comporte facultativement l'attribut "définition de modèle" (*template definition*) qui contient une représentation textuelle de la définition du modèle GDMO.

La définition de la classe d'Annuaire auxiliaire modèles de classes d'objets gérés (*managed object class template*) la place dans les entrées d'Annuaire de la classe d'Annuaire structurelle informations enregistrées. Cette classe représente les spécifications GDMO des classes d'objets gérés. Elle comporte obligatoirement les attributs "dérivée de" (*derived from*), "caractérisée par" (*characterized by*), "blocs de propriétés conditionnels" (*conditional packages*) et "enregistrée sous" (*registered as*).

L'attribut "dérivée de" identifie les hyperclasses immédiates d'une classe d'objets gérés. L'attribut "caractérisée par" spécifie les blocs de propriétés obligatoires d'une classe d'objets gérés tandis que l'attribut "blocs de propriétés conditionnels" identifie ces blocs et leurs conditions d'usage. L'attribut "enregistrée sous" contient l'identificateur d'objet de la classe d'objets gérés représentée.

La définition de la classe d'Annuaire auxiliaire modèles de blocs de propriétés (*package template*) la place dans les entrées d'Annuaire de la classe d'Annuaire structurelle informations enregistrées. Cette classe représente les spécifications GDMO des blocs de propriétés. Elle comporte obligatoirement les attributs "comportement" (*behaviour*), "attributs" (*attributes*), "groupes d'attributs" (*attribute groups*), "actions" (*actions*), "notifications" (*notifications*) et "facultativement enregistré sous" (*optionally registered as*).

L'attribut "comportement" spécifie le comportement associé au bloc de propriétés. L'attribut "attributs" identifie les attributs, ainsi que les qualificatifs et paramètres associés, du bloc de propriétés représenté. L'attribut "groupes d'attributs" identifie les groupes d'attributs, ainsi que les attributs associés, du bloc de propriétés représenté. L'attribut "actions" identifie les actions, ainsi que les paramètres associés, du bloc de propriétés représenté. L'attribut "notifications" identifie les notifications, ainsi que les paramètres associés, du bloc de propriétés représenté. L'attribut "facultativement enregistré sous" contient soit l'identificateur d'objet du bloc représenté soit, si aucun identificateur n'a été attribué, la valeur "néant" (*null*).

La définition de la classe d'Annuaire auxiliaire modèles de paramètres (*parameter template*) la place dans les entrées d'Annuaire de la classe d'Annuaire structurelle informations enregistrées. Cette classe représente les spécifications GDMO des paramètres. Elle comporte obligatoirement les attributs "contexte" (*context*), "syntaxe ou attribut" (*syntax or attribute*), "comportement" et "facultativement enregistré sous". L'attribut "contexte" identifie le contexte du paramètre identifié. L'attribut "syntaxe ou attribut" identifie la spécification de syntaxe du paramètre représenté. L'attribut "comportement" spécifie le comportement associé au paramètre représenté. L'attribut "facultativement enregistré sous" contient soit l'identificateur d'objet du paramètre représenté soit, si aucun identificateur n'a été attribué, la valeur "néant".

La définition de la classe d'Annuaire auxiliaire modèles de relations de dénomination (*name binding template*) la place dans les entrées d'Annuaire de la classe d'Annuaire structurelle informations enregistrées. Cette classe représente les spécifications GDMO des relations de dénomination. Elle comporte obligatoirement les attributs "classe d'objets gérés subordonnée" (*subordinate managed object class*), "dénommée par la classe d'objets gérés supérieure" (*named by superior object class*), "avec l'attribut" (*with attribute*), "comportement" (*behaviour*), "création" (*create*), "suppression" (*delete*) et "enregistrée sous" (*registered as*).

L'attribut "classe d'objets gérés subordonnée" identifie les classes d'objets gérés subordonnées de la relation de dénomination représentée. L'attribut "dénommée par la classe d'objets gérés supérieure" identifie les classes d'objets supérieures de la relation de dénomination représentée. L'attribut "avec l'attribut" identifie l'attribut de dénomination de la relation de dénomination représentée. L'attribut "comportement" spécifie le comportement associé à la relation de dénomination représentée. Les attributs "création" et "suppression" contiennent les spécifications des éléments création et suppression de la relation de dénomination représentée. Au cas où l'un des éléments création ou suppression ne serait pas présent, l'attribut correspondant prendrait obligatoirement la valeur "néant". L'attribut "enregistrée sous" contient l'identificateur d'objet de la relation de dénomination représentée.

La définition de la classe d'Annuaire auxiliaire modèles d'attributs (*attribute template*) la place dans les entrées d'Annuaire de la classe d'Annuaire structurelle informations enregistrées. Cette classe représente les spécifications GDMO des attributs. Elle comporte obligatoirement les attributs "dérivé ou avec choix de syntaxe" (*derived or with syntax choice*), "concorde si" (*matches for*), "comportement", "paramètres" et "facultativement enregistré sous". L'attribut "dérivé ou avec choix de syntaxe" contient la spécification de syntaxe de l'attribut représenté. L'attribut "concorde si" spécifie les caractéristiques de concordance de l'attribut représenté. L'attribut "comportement" spécifie le comportement associé à l'attribut représenté. L'attribut "paramètres" identifie les paramètres associés à l'attribut représenté. L'attribut "facultativement enregistré sous" contient soit l'identificateur d'objet de l'attribut représenté soit, si aucun identificateur n'a été attribué, la valeur "néant".

La définition de la classe d'Annuaire auxiliaire modèles de groupes d'attributs (*attribute group template*) la place dans les entrées d'Annuaire de la classe d'Annuaire structurelle informations enregistrées. Cette classe représente les spécifications GDMO des groupes d'attributs. Elle comporte obligatoirement les attributs "éléments du groupe" (*group elements*), "figé" (*fixed*), "description" (*description*) et "enregistré sous" (*registered as*). L'attribut "éléments du groupe" spécifie les éléments du groupe d'attributs représenté. L'attribut "figé" détermine si le groupe d'attributs représenté est figé ou dynamique. L'attribut "description" contient la description du groupe d'attributs représenté. L'attribut "enregistré sous" contient l'identificateur d'objet du groupe d'attributs représenté.

La définition de la classe d'Annuaire auxiliaire modèles de comportement (*behaviour template*) la place dans les entrées d'Annuaire de la classe d'Annuaire structurelle informations enregistrées. Cette classe représente les spécifications GDMO de comportement. Elle comporte obligatoirement l'attribut "défini comme" (*defined as*) qui spécifie la définition du comportement.

La définition de la classe d'Annuaire auxiliaire modèles d'actions (*action template*) la place dans les entrées d'Annuaire de la classe d'Annuaire structurelle informations enregistrées. Cette classe représente les spécifications GDMO d'actions. Elle comporte obligatoirement les attributs "comportement", "mode confirmé" (*mode confirmed*), "paramètres", "avec syntaxe d'informations" (*with information syntax*), avec syntaxe de réponse (*with reply syntax*) et "enregistrée sous".

L'attribut "comportement" spécifie le comportement associé à l'action représentée. L'attribut "mode confirmé" détermine si l'action représentée est confirmée ou non. L'attribut "paramètres" identifie les paramètres associés à l'action représentée. L'attribut "avec syntaxe d'informations" spécifie la syntaxe de l'argument de l'action tandis que l'attribut "avec syntaxe de réponse" contient la syntaxe de la réponse à l'action. L'attribut "enregistrée sous" contient l'identificateur d'objet de l'action représentée.

La définition de la classe d'Annuaire auxiliaire modèles de notifications (*notification template*) la place dans les entrées d'Annuaire de la classe d'Annuaire structurelle informations enregistrées. Cette classe représente les spécifications GDMO de notifications. Elle comporte obligatoirement les attributs "comportement", "mode confirmé", "paramètres", "avec syntaxe d'informations", "et identificateurs d'attributs" (*and attribute Id's*), "avec syntaxe de réponse" et "enregistrée sous".

L'attribut "comportement" spécifie le comportement associé à la notification représentée. L'attribut "mode confirmé" détermine si la notification représentée est confirmée ou non. L'attribut "paramètres" identifie les paramètres associés à la notification représentée. L'attribut "avec syntaxe d'informations" spécifie la syntaxe de l'argument de la notification tandis que l'attribut "avec syntaxe de réponse" contient la syntaxe de la réponse à la notification. L'attribut "et identificateurs d'attributs" spécifie les affectations d'attributs aux champs de l'argument de la notification. L'attribut "enregistrée sous" contient l'identificateur d'objet de la notification représentée.

La définition de la classe d'Annuaire auxiliaire modules ASN.1 (*ASN.1 module*) la place dans les entrées d'Annuaire de la classe d'Annuaire structurelle informations enregistrées. Cette classe représente le contenu d'un module ASN.1. Elle comporte obligatoirement les attributs "référence du module" (*module reference*), "contenu du module ASN.1" (*ASN.1 module contents*), "facultativement enregistré sous" et "version d'ASN.1" (*ASN.1 version*). L'attribut "référence du module" identifie le nom du module ASN.1. L'attribut "contenu du module ASN.1" contient le contenu du module ASN.1 sous forme textuelle. L'attribut "version d'ASN.1" identifie la version, ou les versions, de compilateurs ASN.1 qui devraient être capables d'analyser ce module. L'attribut "facultativement enregistré sous" contient soit l'identificateur d'objet du module ASN.1 représenté soit, si aucun identificateur n'a été attribué, la valeur "néant".

9 Définition du service

9.1 Introduction

La fonction de connaissance de gestion offre des services d'action.

9.2 Objets gérés de connaissance de gestion

Le service PT-GET donne le moyen d'extraire les informations relatives aux objets gérés de connaissance de gestion.

9.3 Services de notification

La présente Recommandation | Norme internationale ne définit pas de notifications.

9.4 Services d'action

9.4.1 Définition du service d'obtention de représentation textuelle

Le service d'obtention de représentation textuelle (*get textual representation*) s'applique sur le service M-ACTION de CMIS de la manière indiquée ci-dessous.

Il utilise les paramètres définis au 8.3.1 en plus des paramètres généraux du service M-ACTION définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595. Le Tableau 3 présente la liste des paramètres.

Tableau 3 – Service d'obtention de représentation textuelle

Nom de paramètre	dem./ind.	rép./conf.
Identificateur de demande	P	P
Identificateur de corrélation	–	P
Mode	P	–
Classe d'objets de base	P	–
Instance d'objet de base	P	–
Domaine d'application	P	–
Filtre	P	–
Classe d'objets gérés	–	P
Instance d'objet géré	–	P
Contrôle d'accès	P	–
Synchronisation	P	–
Type d'action	M	C(=)
Informations d'action		
Identificateur d'attribut	M	–
Résultats d'action	–	M
Représentation textuelle	–	M
Attribut non disponible par cette action	–	U
Erreurs	–	U (Note)
NOTE – Si la réponse ne tient pas dans une seule PDU (si, par exemple, il n'est pas possible d'utiliser des réponses corrélées), l'indication d'erreur de CMIP limitation de complexité (<i>ComplexityLimitation</i>) pourra être renvoyée.		

9.4.2 Définition du service de recherche d'arbre d'informations de gestion

Le service de recherche d'arbre d'informations de gestion (*management information tree search*) s'applique sur le service M-ACTION de CMIS de la manière indiquée ci-dessous.

Il utilise les paramètres définis au 8.3.2 en plus des paramètres généraux du service M-ACTION général définis dans la Rec. X.710 du CCITT et l'ISO/CEI 9595. Le Tableau 4 présente la liste des paramètres.

9.5 Services d'Annuaire

Les opérations de lecture et de recherche de l'Annuaire donnent le moyen d'extraire les informations relatives aux objets gérés de connaissance de gestion.

Tableau 4 – Service de recherche d'arbre d'informations de gestion

Nom de paramètre	dem./ind.	rép./conf.
Identificateur de demande	P	P
Identificateur de corrélation	–	P
Mode	P	–
Classe d'objets de base	P	–
Instance d'objet de base	P	–
Domaine d'application	P	–
Filtre	P	–
Classe d'objets gérés	–	P
Instance d'objet géré	–	P
Contrôle d'accès	P	–
Synchronisation	P	–
Type d'action	M	C(=)
Informations d'action		
Base	M	–
Domaine d'exploration	M	–
Demande de classe	U	–
Résultats d'action	–	M
Arbre de dénomination	–	M
Objet racine	–	M
Classe de la racine	–	C (Note 1)
Subordonnés	–	U
RDN	–	M
Classe d'objets gérés (pour subordonnés)	–	C (Note 1)
Erreurs	–	U (Note 2)
NOTES		
1 La classe d'objets gérés de la racine de l'arbre de dénomination et celle de chaque subordonné dans cet arbre doivent être spécifiées si le paramètre de demande de classe a la valeur «Vrai» dans la demande de service de recherche d'arbre d'informations de gestion; sinon, la classe d'objets gérés de la racine de l'arbre de dénomination et celle de chaque subordonné seront absentes de la réponse.		
2 Si la réponse ne tient pas dans une seule PDU (si, par exemple, le domaine d'exploration est étendu et si des réponses corrélées ne peuvent pas être utilisées), l'indication d'erreur de CMIP limitation de complexité (<i>ComplexityLimitation</i>) pourra être renvoyée.		

10 Unités fonctionnelles de gestion-systèmes

La présente Recommandation | Norme internationale ne spécifie pas d'unités fonctionnelles. Les classes d'objets gérés qu'elle définit spécifient comment un système géré peut faire connaître le répertoire de ses capacités, ses définitions d'informations de gestion et ses objets gérés.

Afin de gérer la connaissance de la gestion, il peut être fait appel aux unités fonctionnelles négociables suivantes de la Rec. X.730 du CCITT | ISO/CEI 10164-1:

- commande (*control*);
- contrôle (*monitor*);
- événements sur objets (*objectEvents*).

11 Protocole et syntaxe abstraite

11.1 Syntaxe abstraite

11.1.1 Objets gérés

11.1.1.1 Objets gérés référencés

La présente Recommandation | Norme internationale fait référence aux objets gérés ci-dessous. Leur syntaxe abstraite est spécifiée dans la Rec. X.721 du CCITT | ISO/CEI 10165-2:

- a) sommet (*top*);
- b) système (*system*).

11.1.1.2 Objets gérés définis

Les Tableaux 5, 6 et 7 montrent la relation qui existe entre les objets gérés définis au 8.1 et les spécifications d'étiquettes de référence des modèles d'objets gérés de l'Annexe A.

Tableau 5 – Objets gérés pour la connaissance des répertoires

Nom de classe d'objets gérés	Etiquette de référence
Répertoire de classe d'objets gérés	managedObjectClassRepertoire
Répertoire	repertoire

Tableau 6 – Objets gérés pour la connaissance des définitions

Nom de classe d'objets gérés	Etiquette de référence
Modèle d'action	actionTemplate
Module ASN.1	ASN.1Module
Modèle de groupe d'attributs	attributeGroupTemplate
Modèle d'attribut	attributeTemplate
Modèle de comportement	behaviourTemplate
Document	document
Modèle de classe d'objets gérés	managedObjectClassTemplate
Modèle de relation de dénomination	nameBindingTemplate
Modèle de notification	notificationTemplate
Modèle de bloc de propriétés	packageTemplate
Modèle de paramètre	parameterTemplate
Modèle	template

Tableau 7 – Objets gérés pour la connaissance des instances

Nom de classe d'objets gérés	Etiquette de référence
Objet géré exploration	discoveryObject

11.1.2 Attributs

11.1.2.1 Attributs définis dans la présente Recommandation | Norme internationale

Les Tableaux 8, 9 et 10 montrent la relation qui existe entre les attributs définis au 8.1 et les spécifications d'étiquettes de référence des modèles d'attributs de l'Annexe A.

Tableau 8 – Attributs pour la connaissance des répertoires

Nom d'attribut	Etiquette de référence
Valeurs implémentées	implementedValues
Identificateur de répertoire de classes d'objets gérés	managedObjectClassRepertoireId
Identificateur de répertoire	repertoireId
Profils de CMIP mis en œuvre	supportedCmipProfiles
Liste de blocs de propriétés conditionnels mis en œuvre	supportedConditionalPackageList
Liste de classes d'objets gérés mises en œuvre	supportedManagedObjectClassList
Liste de relations de dénomination mises en œuvre	supportedNameBindingList
Liste de classes de relations mises en œuvre	supportedRelationshipClassList

11.1.3 Actions

Les Tableaux 11 et 12 montrent la relation qui existe entre les actions définies au 8.3 et les spécifications d'étiquettes de référence d'actions de l'Annexe A.

11.1.4 Notifications

11.1.4.1 Notifications référencées

La présente Recommandation | Norme internationale fait référence aux événements spécifiés dans la Rec. X.730 du CCITT | ISO/CEI 10164-1:

- a) notification de modification de valeur d'attribut (*attribute value change notification*);
- b) notification de création d'objet (*object creation notification*);
- c) notification de suppression d'objet (*object deletion notification*).

11.1.4.2 Notifications définies dans la présente Recommandation | Norme internationale

La présente Recommandation | Norme internationale ne définit aucune notification.

11.1.5 Classes d'objets d'Annuaire

11.1.5.1 Classes d'objets d'Annuaire référencées

La présente Recommandation | Norme internationale fait référence aux deux classes d'objets d'Annuaire ci-dessous. Leurs syntaxes abstraites sont spécifiées respectivement dans la Rec. UIT-T X.501 | ISO/CEI 9594-2 et dans la Rec. UIT-T X.521 | ISO/CEI 9594-7:

- a) sommet (*top*);
- b) entité d'application (*applicationEntity*).

Tableau 9 – Attributs pour la connaissance des définitions

Nom d'attribut	Etiquette de référence
Actions	actions
Et identificateurs d'attribut	andAttributeIds
Contenu de module ASN.1	asn1ModuleContents
Version d'ASN.1	asn1Version
Groupes d'attributs	attributeGroups
Attributs	attributes
Comportement	behaviour
Caractérisé par	characterizedBy
Blocs de propriétés conditionnels	conditionalPackages
Contexte	context

Tableau 9 – Attributs pour la connaissance des définitions (fin)

Nom d'attribut	Etiquette de référence
Création	create
Défini comme	definedAs
Suppression	delete
Dérivée de	derivedFrom
Dérivé ou avec choix de syntaxe	derivedOrWithSyntaxChoice
Description	description
Nom de document	documentName
Identificateur d'objet de document	documentObjectIdentifier
Figé	fixed
Éléments du groupe	groupElements
Concorde si	matchesFor
Mode confirmé	modeConfirmed
Référence du module	moduleReference
Dénommée par classe d'objets supérieure	namedBySuperiorObjectClass
Notifications	notifications
Facultativement enregistré(e) sous	optionallyRegisteredAs
Paramètres	parameters
Enregistré(e) sous	registeredAs
Spécification	specification
Classe d'objets gérés subordonnée	subordinateObjectClass
Syntaxe ou attribut	syntaxOrAttribute
Définition de modèle	templateDefinition
Nom de modèle	templateName
Avec l'attribut	withAttribute
Avec la syntaxe d'informations	withInformationSyntax
Avec la syntaxe de réponse	withReplySyntax

Tableau 10 – Attributs pour la connaissance des instances

Nom d'attribut	Etiquette de référence
Identificateur d'exploration	discoveryId

Tableau 11 – Actions pour la connaissance des définitions

Nom de l'action	Etiquette de référence
Obtention de représentation textuelle	getTextualRepresentation

Tableau 12 – Actions pour la connaissance des instances

Nom de l'action	Etiquette de référence
Recherche d'arbre d'informations de gestion	mITSearch

11.1.5.2 Classes d'objets d'Annuaire définies

Les Tableaux 13 et 14 montrent la relation qui existe entre les classes d'objets d'Annuaire définies au 8.6 et les spécifications d'étiquettes de référence des classes d'objets d'Annuaire de l'Annexe B.

Tableau 13 – Objets d'Annuaire pour la connaissance des répertoires

Nom de classe d'objets d'Annuaire	Etiquette de référence
Elément de service commun d'informations de gestion	cMISE
Elément de service d'application de gestion-systèmes	sMASE

Tableau 14 – Objets d'Annuaire pour la connaissance des définitions

Nom de classe d'objets d'Annuaire	Etiquette de référence
Module ASN.1	asn1Module
Modèle d'action	managementAction
Modèle d'attribut	managementAttribute
Modèle de groupe d'attribut	managementAttributeGroup
Modèle de comportement	managementBehaviour
Document	managementDocument
Modèle de relation de dénomination	managementNameBinding
Modèle de notification	managementNotification
Modèle de classe d'objets gérés	managementObjectClass
Modèle de bloc de propriétés	managementPackage
Modèle de paramètre	managementParameter
Modèle	managementTemplate
Informations enregistrées	registeredInformation

Le Tableau 15 présente les classes d'objets d'Annuaire auxiliaires qu'il est nécessaire d'inclure dans les entrées de la classe d'Annuaire informations enregistrées et les classes d'objets gérés de définition associées.

11.1.6 Attributs d'Annuaire

11.1.6.1 Attributs d'Annuaire référencés

La présente Recommandation | Norme internationale fait référence aux attributs d'Annuaire dont la syntaxe abstraite est spécifiée dans la Rec. UIT-T X.520 | ISO/CEI 9594-6.

Tableau 15 – Classes d'objets gérés de définition et classes d'Annuaire auxiliaires associées

Etiquette de classe d'objets gérés	Etiquette de classe d'Annuaire auxiliaire
actionTemplate	managementAction
aSN1Module	asn1Module
attributeGroupTemplate	managementAttributeGroup
attributeTemplate	managementAttribute
behaviourTemplate	managementBehaviour
document	managementDocument
managedObjectClassTemplate	managementObjectClass
nameBindingTemplate	managementNameBinding
notificationTemplate	managementNotification
packageTemplate	managementPackage
parameterTemplate	managementParameter
template	managementTemplate

11.1.6.2 Attributs d'Annuaire définis dans la présente Recommandation | Norme internationale

Les Tableaux 16 et 17 montrent la relation qui existe entre les attributs d'Annuaire définis au 8.6 et les spécifications d'étiquettes de référence des attributs de l'Annexe B.

Tableau 16 – Attributs d'Annuaire pour la connaissance des répertoires

Nom d'attribut d'Annuaire	Etiquette de référence
Liste des objets gérés d'arbre d'informations de gestion	mitMoList
Profils de CMIP mis en œuvre	supportedCmipProfiles
Version de CMIP mise en œuvre	supportedCmipVersion
Unités fonctionnelles de CMIS mises en œuvre	supportedCmisFunctionalUnits
Unités fonctionnelles de SMASE mises en œuvre	supportedSmaseFunctionalUnits
Met en œuvre les noms globaux de MKM	supportsMKMGlobalNames

11.2 Éléments de procédure pour l'obtention de représentation textuelle

11.2.1 Rôle de gestionnaire

11.2.1.1 Demande

Les procédures d'obtention de représentation textuelle sont déclenchées par la primitive de service de demande d'obtention de représentation textuelle. A la réception de cette primitive, la machine protocolaire SMAPM construit une unité MAPDU et émet une primitive de demande M-ACTION du service CMIS dont les paramètres sont déduits de ceux de la primitive de demande d'obtention de représentation textuelle.

11.2.1.2 Réception de la réponse

A la réception d'une primitive de confirmation M-ACTION du service CMIS contenant une MAPDU qui répond à une action d'obtention de représentation textuelle et qui contient l'identificateur de corrélation, la SMAPM émet une primitive de confirmation d'obtention de représentation textuelle vers l'utilisateur de ce service. Les paramètres de cette primitive sont déduits de ceux de la primitive de confirmation M-ACTION du service CMIS. La SMAPM indique que la procédure n'est pas terminée.

A la réception d'une primitive de confirmation M-ACTION du service CMIS contenant une MAPDU qui répond à une action d'obtention de représentation textuelle et qui *ne contient pas* l'identificateur de corrélation, la SMAPM émet une primitive de confirmation d'obtention de représentation textuelle vers l'utilisateur de ce service. Les paramètres de cette primitive de confirmation sont déduits de ceux de la primitive de confirmation M-ACTION du service CMIS. La SMAPM indique que la procédure est terminée.

11.2.2 Rôle d'agent

11.2.2.1 Réception d'une demande

A la réception d'une primitive d'indication M-ACTION du service CMIS contenant une MAPDU bien formée qui demande le service d'obtention de représentation textuelle, la SMAPM doit émettre une primitive d'indication d'obtention de représentation textuelle vers l'utilisateur de ce service. Les paramètres de cette indication sont dérivés de ceux de la primitive d'indication M-ACTION du service CMIS. Si la MAPDU reçue n'est pas bien formée, la SMAPM doit construire une MAPDU adéquate comportant notification de l'erreur et émettre une primitive de réponse M-ACTION du service CMIS avec présence d'un paramètre d'indication d'erreur.

11.2.2.2 Réponse

S'il est nécessaire d'émettre des réponses corrélées en réponse à une demande de service d'obtention de représentation textuelle, pour toutes les réponses sauf la dernière, la SMAPM doit accepter de recevoir une primitive de réponse d'obtention de représentation textuelle. Elle doit construire une MAPDU contenant une réponse indiquant le succès de l'action et émettre une primitive de réponse M-ACTION du service CMIS dont les paramètres sont dérivés de ceux de la primitive de réponse d'obtention de représentation textuelle. La primitive de réponse à M-ACTION doit contenir l'identificateur de corrélation.

Tableau 17 – Attributs d'Annuaire pour la connaissance des définitions

Nom d'attribut d'Annuaire	Etiquette de référence
Actions	actions
Informations complémentaires	additionalInformation
Et identificateurs d'attribut	andAttributeIds
Contenu de module ASN.1	asn1ModuleContents
Version d'ASN.1	asn1Version
Groupes d'attributs	attributeGroups
Attributs	attributes
Comportement	behaviour
Caractérisé par	characterizedBy
Blocs de propriétés conditionnels	conditionalPackages
Contexte	context
Création	create
Défini comme	definedAs
Suppression	delete
Dérivée de	derivedFrom
Dérivé ou avec choix de syntaxe	derivedOrWithSyntaxChoice
Description	description
Nom de document	documentName
Identificateur d'objet de document	documentObjectIdentifier
Fixé	fixed
Eléments du groupe	groupElements
Statut d'information	informationStatus
Concorde si	matchesFor
Mode confirmé	modeConfirmed
Référence de module	moduleReference
Dénommée par classe d'objets supérieure	namedBySuperiorObjectClass
Forme de nom	nameForm
Notifications	notifications
Facultativement enregistré(e) sous	optionallyRegisteredAs
Paramètres	parameters
Enregistré(e) sous	registeredAs
Spécification	specification
Classe d'objets gérés subordonnée	subordinateObjectClass
Syntaxe ou attribut	syntaxOrAttribute
Définition de modèle	templateDefinition
Nom de modèle	templateName
Avec l'attribut	withAttribute
Avec la syntaxe d'informations	withInformationSyntax
Avec la syntaxe de réponse	withReplySyntax

La SMAPM doit accepter de recevoir une primitive de réponse d'obtention de représentation textuelle. Elle doit construire une MAPDU contenant une réponse indiquant soit le succès, soit une erreur, de l'action et émettre une primitive de réponse M-ACTION du service CMIS dont les paramètres sont dérivés de ceux de la primitive de réponse d'obtention de représentation textuelle. La primitive de réponse à M-ACTION ne doit pas contenir l'identificateur de corrélation.

NOTE – Si des réponses multiples sont émises, la dernière ne contient que l'identificateur de demande.

11.3 Eléments de procédure pour la recherche d'arbre d'informations de gestion

11.3.1 Rôle de gestionnaire

11.3.1.1 Demande

Les procédures de recherche d'arbre d'informations de gestion sont déclenchées par la primitive de service de demande de recherche d'arbre d'informations de gestion. A la réception de cette primitive, la machine protocolaire SMAPM construit une MAPDU et émet une primitive de demande M-ACTION du service CMIS dont les paramètres sont déduits de ceux de la primitive de demande de recherche d'arbre d'informations de gestion.

11.3.1.2 Réception de la réponse

A la réception d'une primitive de confirmation M-ACTION du service CMIS contenant une MAPDU qui répond à une action de recherche d'arbre d'informations de gestion, la SMAPM émet une primitive de confirmation de recherche d'arbre d'informations de gestion vers l'utilisateur de ce service. Les paramètres de cette primitive sont déduits de ceux de la primitive de confirmation M-ACTION du service CMIS. La procédure est alors terminée.

11.3.2 Rôle d'agent

11.3.2.1 Réception d'une demande

A la réception d'une primitive d'indication M-ACTION du service CMIS bien formée contenant une MAPDU qui demande le service de recherche d'arbre d'informations de gestion, la SMAPM doit émettre une primitive d'indication de recherche d'arbre d'informations de gestion vers l'utilisateur de ce service. Les paramètres de cette indication sont dérivés de ceux de la primitive d'indication M-ACTION du service CMIS. Si la MAPDU reçue n'est pas bien formée, la SMAPM doit construire une MAPDU adéquate comportant notification de l'erreur et émettre une primitive de réponse M-ACTION du service CMIS avec présence d'un paramètre d'indication d'erreur.

11.3.2.2 Réponse

La SMAPM doit accepter de recevoir une primitive de réponse de recherche d'arbre d'informations de gestion. Elle doit construire une MAPDU contenant une réponse indiquant soit le succès, soit une erreur, de l'action et émettre une primitive de réponse M-ACTION du service CMIS dont les paramètres sont dérivés de ceux de la primitive de service de réponse de recherche d'arbre d'informations de gestion.

11.4 Négociation des unités fonctionnelles

La présente Recommandation | Norme internationale ne spécifie pas d'unités fonctionnelles.

11.5 Eléments caractéristiques du protocole CMIP

Si le protocole CMIP est utilisé, la présente fonction exige la mise en œuvre du nom distinctif local.

12 Relations avec d'autres fonctions et avec l'Annuaire

La fonction de gestion de la connaissance de gestion emploie les services définis dans la Rec. X.730 du CCITT | ISO/CEI 10164-1 pour la création et la suppression des objets gérés de connaissance de gestion, l'extraction de leurs attributs et pour les notifications de création et de suppression d'objets et de modification de valeurs d'attributs.

Les mécanismes de commande d'accès à employer avec les objets gérés de gestion de connaissance de gestion sont spécifiés dans la Rec. UIT-T X.741 | ISO/CEI 10164-9.

La fonction de gestion de la connaissance de gestion emploie les services définis dans la Rec. UIT-T X.511 | ISO/CEI 9594-3 pour la création et la suppression des objets d'Annuaire de connaissance de gestion et pour l'extraction de leurs attributs.

Les mécanismes de commande d'accès à employer avec les objets d'Annuaire de gestion de la connaissance de gestion sont spécifiés dans la Rec. UIT-T X.501 | ISO/CEI 9594-2.

13 Conformité

Les implémentations qui se déclarent conformes à la présente Recommandation | Norme internationale doivent respecter les exigences de conformité telles qu'elles sont spécifiées dans les paragraphes suivants.

13.1 Conformité statique

L'implémentation doit se conformer aux exigences de la présente Recommandation | Norme internationale dans l'un au moins des rôles suivants:

- rôle de gestionnaire;
- rôle d'agent;
- rôle d'utilisateur d'informations d'Annuaire;
- rôle de fournisseur d'informations d'Annuaire.

Une déclaration de conformité à l'un au moins de ces rôles doit être faite selon le Tableau C.1.

En cas de déclaration de conformité pour la mise en œuvre du rôle de gestionnaire, l'implémentation doit mettre en œuvre au moins une des opérations de gestion ou de notification pour l'un au moins des objets gérés spécifiés dans la présente Recommandation | Norme internationale. Les exigences de conformité dans le rôle de gestionnaire relatives à ces opérations ou notifications sont identifiées dans le Tableau C.2 et dans les autres tableaux référencés dans l'Annexe C.

En cas de déclaration de conformité pour la mise en œuvre du rôle d'agent, l'implémentation doit mettre en œuvre une ou plusieurs instances des classes d'objets gérés identifiées dans le Tableau C.3 et dans les autres tableaux référencés dans l'Annexe C.

En cas de déclaration de conformité pour la mise en œuvre du rôle d'utilisateur d'informations d'Annuaire, l'implémentation doit contenir un agent DUA capable d'extraire les informations appartenant aux objets d'Annuaire de répertoire ou aux objets d'Annuaire de définition.

En cas de déclaration de conformité pour la mise en œuvre du rôle de fournisseur d'informations d'Annuaire, l'implémentation doit contenir un agent DSA capable de fournir les informations appartenant aux objets d'Annuaire de répertoire ou aux objets d'Annuaire de définition.

L'implémentation doit mettre en œuvre la syntaxe de transfert dérivée des règles de codage spécifiées dans la Rec. X.209 du CCITT et l'ISO/CEI 8825 dénommée {joint-iso-ccitt(2) asn1(1) basicEncoding (1)} pour les types de données abstraites référencés par les définitions dont la mise en œuvre est annoncée.

13.2 Conformité dynamique

Les implémentations qui se déclarent conformes à la présente Recommandation | Norme internationale doivent mettre en œuvre les éléments de procédure et les définitions sémantiques correspondant aux définitions dont la mise en œuvre est annoncée.

13.3 Exigences relatives aux déclarations de conformité d'une implémentation de la gestion

Tout formulaire MCS, MICS, MOCS ou MRCS qui se conforme à la présente Recommandation | Norme internationale doit être techniquement identique aux formulaires spécifiés aux Annexes C, D, E et F, n'en différant que dans les limites permises par la Rec. UIT-T X.724 | ISO/CEI 10165-6.

Le fournisseur d'une implémentation présentée comme conforme à la présente Recommandation | Norme internationale doit renseigner, au titre des exigences de conformité, une copie du formulaire MCS fourni dans l'Annexe C, ainsi qu'une copie de tout autre formulaire ICS que ce MCS désigne comme étant applicable. Une déclaration ICS qui se conforme à la présente Recommandation | Norme internationale doit:

- décrire une implémentation conforme à la présente Recommandation | Norme internationale;
- avoir été renseignée dans le respect des instructions données dans la Rec. UIT-T X.724 | ISO/CEI 10165-6;
- inclure les informations nécessaires pour pouvoir identifier sans ambiguïté et le fournisseur et l'implémentation.

Annexe A

Objets gérés de connaissance de gestion

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

A.1 Allocation of object identifiers

This Recommendation | International Standard allocates the following object identifiers.

```
--%PRAGMA version BIT STRING {v1990(0), v1994(1)} ::= {v1990, v1994}
MKMD {joint-iso-ccitt(2) ms(9) function(2) part16(16) asn1Modules(2) 5}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
dmiAttribute                OBJECT IDENTIFIER ::=
    {joint-iso-ccitt(2) ms(9) smi(3) part2(2) attribute(7)}
dmiNotification            OBJECT IDENTIFIER ::=
    {joint-iso-ccitt(2) ms(9) smi(3) part2(2) notification(10)}
mkmDirectoryAttributeType  OBJECT IDENTIFIER ::=
    {joint-iso-ccitt(2) ms(9) function(2) part16(16) standardSpecificExtension(0)
    directoryAttributeTypes(4)}
mkmDirectoryObjectClass    OBJECT IDENTIFIER ::=
    {joint-iso-ccitt(2) ms(9) function(2) part16(16) standardSpecificExtension(0) directoryObjectClasses(6)}
mkmDirectoryNameForm       OBJECT IDENTIFIER ::=
    {joint-iso-ccitt(2) ms(9) function(2) part16(16) standardSpecificExtension(0) directoryNameForms(7)}
mkmMObjectClass            OBJECT IDENTIFIER ::=
    {joint-iso-ccitt(2) ms(9) function(2) part16(16) managedObjectClass(3)}
mkmPackage                 OBJECT IDENTIFIER ::=
    {joint-iso-ccitt(2) ms(9) function(2) part16(16) package(4)}
mkmParameter              OBJECT IDENTIFIER ::=
    {joint-iso-ccitt(2) ms(9) function(2) part16(16) parameter(5)}
mkmNameBinding             OBJECT IDENTIFIER ::=
    {joint-iso-ccitt(2) ms(9) function(2) part16(16) nameBinding(6)}
mkmAttribute              OBJECT IDENTIFIER ::=
    {joint-iso-ccitt(2) ms(9) function(2) part16(16) attribute(7)}
mkmAction                 OBJECT IDENTIFIER ::=
    {joint-iso-ccitt(2) ms(9) function(2) part16(16) action(9)}
END
```

A.2 Repertoire managed object definitions

A.2.1 Managed object classes

```
managedObjectClassRepertoire MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":top;
CHARACTERIZED BY managedObjectClassRepertoirePackage PACKAGE
BEHAVIOUR managedObjectClassRepertoireBehaviour BEHAVIOUR DEFINED AS !
    Managed objects with this behaviour issue the object creation notification upon creation, the object deletion
    notification upon deletion and the attribute value change notification upon a change of value of either the
    supported conditional package list attribute or the implemented values attribute.
    !;;
ATTRIBUTES
    managedObjectClassRepertoireId GET,
    supportedConditionalPackageList GET;
NOTIFICATIONS
    "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":attributeValueChange,
    "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":objectCreation,
    "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":objectDeletion;;;
CONDITIONAL PACKAGES
    implementedValuesPackage PACKAGE
```

BEHAVIOUR implementedValuesPackageBehaviour BEHAVIOUR DEFINED AS !

Presence of the implemented values conditional package implies that the managed object implementation supports the indicated implemented values for each of the managed object attributes indicated within the implemented values attribute.

!;;

ATTRIBUTES

implementedValues GET;

REGISTERED AS {MKMD.mkmPackage 2}; PRESENT IF "an instance supports it";

REGISTERED AS {MKMD.mkmMObjectClass 2};

repertoire MANAGED OBJECT CLASS

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

CHARACTERIZED BY repertoirePackage PACKAGE

BEHAVIOUR repertoireBehaviour BEHAVIOUR DEFINED AS !

An instance with this behaviour reflects system capabilities: when a system is enabled for support of new managed object classes, new managed relationship classes, new name bindings, or new CMIP profiles, or when existing support is disabled, the new capability shall be reflected in the values of the corresponding attributes.

An instance with this behaviour shall emit the object creation notification upon creation, the object deletion notification upon deletion and the attribute value change notification upon a change of value of the supported managed object class list, supported relationship class list, supported name binding list, or supported CMIP profiles attribute.

If a managed system has any instances of the repertoire managed object, it shall have one named under the system managed object with the value "REP1" for the repertoire Id attribute. A managed system may have multiple instances of the repertoire managed object.

!;;

ATTRIBUTES

repertoireId GET,

supportedNameBindingList GET,

supportedManagedObjectClassList GET,

supportedRelationshipClassList GET;

NOTIFICATIONS

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":objectCreation,

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":objectDeletion,

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":attributeValueChange;;;

CONDITIONAL PACKAGES

supportedCmipProfilesPackage PACKAGE

ATTRIBUTES

supportedCmipProfiles GET;

REGISTERED AS {MKMD.mkmPackage 1}; PRESENT IF "this managed object represents the managed system components that support communications";

REGISTERED AS {MKMD.mkmMObjectClass 1};

A.2.2 Name bindings

managedObjectClassRepertoire-repertoire NAME BINDING

SUBORDINATE OBJECT CLASS

managedObjectClassRepertoire AND SUBCLASSES;

NAMED BY SUPERIOR OBJECT CLASS

repertoire AND SUBCLASSES;

WITH ATTRIBUTE managedObjectClassRepertoireId;

BEHAVIOUR managedObjectClassRepertoire-repertoireBehaviour BEHAVIOUR DEFINED AS !

The value of the managed object class repertoire Id attribute shall be the OBJECT IDENTIFIER for the managed object class with which the repertoire is associated.

!;;

REGISTERED AS {MKMD.mkmNameBinding 2};

repertoire-system NAME BINDING

SUBORDINATE OBJECT CLASS

repertoire AND SUBCLASSES;

NAMED BY SUPERIOR OBJECT CLASS

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":system AND SUBCLASSES;

WITH ATTRIBUTE repertoireId;

BEHAVIOUR repertoire-systemBehaviour BEHAVIOUR DEFINED AS !

The use of a fixed relative distinguished name for a common way to access a repertoire managed object is provided by this name binding when the value "REP1" is used for the repertoire Id attribute.

!;;

REGISTERED AS {MKMD.mkmNameBinding 1};

A.2.3 Attributes

implementedValues ATTRIBUTE

WITH ATTRIBUTE SYNTAX RepertoireASN1Module.ImplementedValues;
BEHAVIOUR implementedValuesBehaviour BEHAVIOUR DEFINED AS !

This attribute represents class implementation information related to a set of attributes of a managed object class. This set-valued attribute shall not contain more than one element with the same attributeId value. The information provided applies only to instances of the class for which information is being provided: in other words, implementation information represented by this attribute is not inherited by derived classes.

The following information about an attribute can be represented:

- **initialValue:** The value the attribute first takes when an instance of the class comes into existence.
- **defaultValue:** The value to which the attribute is set when a set-to-default management operation is applied to it.
- **minValue:** For an attribute of type INTEGER, REAL, or a subtype of either of these types, which has ranges for both its permitted and required values sets, minValue indicates the minimum value that the attribute can take in an instance of the class. minValue shall be greater than or equal to the smallest permitted value and less than or equal to the smallest required value.
- **maxValue:** For an attribute of type INTEGER, REAL, or a subtype of either of these types, which has ranges for both its permitted and required values sets, maxValue indicates the maximum value that the attribute can take in an instance of the class. maxValue shall be less than or equal to the largest permitted value and greater than or equal to the largest required value.

!;;

REGISTERED AS {MKMD.mkmAttribute 6};

managedObjectClassRepertoireId ATTRIBUTE

WITH ATTRIBUTE SYNTAX RepertoireASN1Module.ManagedObjectClassRepertoireId;
MATCHES FOR EQUALITY;
BEHAVIOUR managedObjectClassRepertoireIdBehaviour BEHAVIOUR DEFINED AS !

The value of this attribute is the OBJECT IDENTIFIER with which the managed object class was registered.

!;;

REGISTERED AS {MKMD.mkmAttribute 2};

repertoireId ATTRIBUTE

WITH ATTRIBUTE SYNTAX RepertoireASN1Module.RepertoireId;
MATCHES FOR EQUALITY;
BEHAVIOUR repertoireIdBehaviour BEHAVIOUR DEFINED AS !

The repertoireId attribute is a distinguishing attribute suitable for naming.

!;;

REGISTERED AS {MKMD.mkmAttribute 1};

supportedCmipProfiles ATTRIBUTE

WITH ATTRIBUTE SYNTAX RepertoireASN1Module.SupportedCmipProfiles;
MATCHES FOR EQUALITY;
BEHAVIOUR supportedCmipProfilesBehaviour BEHAVIOUR DEFINED AS !

This attribute indicates the CMIP profiles supported by the system operating in the agent role. The following values are defined:

- aom10(0): The managed system supports ISP 11183-1.
- aom11(1): The managed system supports ISP 11183-3.
- aom12(2): The managed system supports ISP 11183-2.

!;;

REGISTERED AS {MKMD.mkmAttribute 4};

supportedConditionalPackageList ATTRIBUTE

WITH ATTRIBUTE SYNTAX RepertoireASN1Module.SupportedConditionalPackageList;
MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
BEHAVIOUR supportedConditionalPackageListBehaviour BEHAVIOUR DEFINED AS !

This set-valued, read-only attribute identifies the conditional packages supported in the implementation of the managed object class.

!;;

REGISTERED AS {MKMD.mkmAttribute 5};

supportedManagedObjectClassList ATTRIBUTE

WITH ATTRIBUTE SYNTAX RepertoireASN1Module.SupportedManagedObjectClassList;
MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
BEHAVIOUR supportedManagedObjectClassListBehaviour BEHAVIOUR DEFINED AS !

The supportedManagedObjectClassList attribute indicates the managed object classes, actual and allomorphic, supported by a system operating in the agent role.

!;;

REGISTERED AS {MKMD.mkmAttribute 3};

supportedNameBindingList ATTRIBUTE

WITH ATTRIBUTE SYNTAX RepertoireASN1Module.SupportedNameBindingList;
MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
BEHAVIOUR supportedNameBindingListBehaviour BEHAVIOUR DEFINED AS !

The supportedNameBindingList attribute indicates the name bindings supported by the system operating in the agent role.

!;;

REGISTERED AS {MKMD.mkmAttribute 42};

supportedRelationshipClassList ATTRIBUTE

WITH ATTRIBUTE SYNTAX RepertoireASN1Module.SupportedRelationshipClassList;
MATCHES FOR SET-COMPARISON, SET-INTERSECTION;
BEHAVIOUR supportedRelationshipClassListBehaviour BEHAVIOUR DEFINED AS !

The supportedRelationshipClassList attribute indicates the relationship classes supported by the system operating in the agent role.

!;;

REGISTERED AS {MKMD.mkmAttribute 44};

A.2.4 Abstract syntax

--%PRAGMA version BIT STRING {v1990(0), v1994(1)} ::= v1990

RepertoireASN1Module {joint-iso-ccitt(2) ms(9) function(2) part16(16) asn1Modules(2) 0}

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- EXPORTS everything;

IMPORTS

NameBinding, Packages

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

-- CCITT Rec. X.721 | ISO/IEC 10165-2

;

GdmoAttributeId ::= OBJECT IDENTIFIER

GdmoObjectClass ::= OBJECT IDENTIFIER

ImplementedValues ::= SET OF SEQUENCE {

attributeId GdmoAttributeId,

defaultValue [1] EXPLICIT ANY DEFINED BY attributeId OPTIONAL,

initialValue [2] EXPLICIT ANY DEFINED BY attributeId OPTIONAL,

minValue [3] EXPLICIT ANY DEFINED BY attributeId OPTIONAL,

maxValue [4] EXPLICIT ANY DEFINED BY attributeId OPTIONAL

}

ManagedObjectClassRepertoireId ::= GdmoObjectClass

RepertoireId ::= GraphicString

SupportedCmipProfiles ::= BIT STRING {

aom10(0), -- the managed system supports ISP 11183-1

aom11(1), -- the managed system supports ISP 11183-3

aom12(2)} -- the managed system supports ISP 11183-2

SupportedConditionalPackageList ::= Packages

SupportedManagedObjectClassList ::= SET OF GdmoObjectClass

SupportedNameBindingList ::= SET OF NameBinding

SupportedRelationshipClassList ::= SET OF OBJECT IDENTIFIER

END

A.3 Definition managed object definitions

A.3.1 Managed object classes

actionTemplate MANAGED OBJECT CLASS

DERIVED FROM template;

CHARACTERIZED BY actionPackage PACKAGE

ATTRIBUTES

behaviour GET,

modeConfirmed GET,

parameters GET,

withInformationSyntax GET,

withReplySyntax GET,

registeredAs GET;;;

REGISTERED AS {MKMD.mkmMOBjectClass 12};

aSN1Module MANAGED OBJECT CLASS**DERIVED FROM** "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":top;**CHARACTERIZED BY** aSN1ModulePackage PACKAGE**BEHAVIOUR** aSN1ModuleBehaviour **BEHAVIOUR DEFINED AS !**

When an instance of this class is named subordinate to a document managed object, the moduleReference attribute provides unique identification within the scope of that document managed object. A managed object with this behaviour supports the retrieval of the asn1ModuleContents attribute via the getTextualRepresentation action.

!;;

ATTRIBUTES

moduleReference GET,
asn1ModuleContents GET,
asn1Version GET,
optionallyRegisteredAs GET;

ACTIONS

getTextualRepresentation;;;

REGISTERED AS {MKMD.mkmMObjectClass 4};**attributeGroupTemplate MANAGED OBJECT CLASS****DERIVED FROM** template;**CHARACTERIZED BY** attributeGroupPackage PACKAGE**ATTRIBUTES**

groupElements GET,
fixed GET,
description GET,
registeredAs GET;;;

REGISTERED AS {MKMD.mkmMObjectClass 10};**attributeTemplate MANAGED OBJECT CLASS****DERIVED FROM** template;**CHARACTERIZED BY** attributePackage PACKAGE**ATTRIBUTES**

derivedOrWithSyntaxChoice GET,
matchesFor GET,
behaviour GET,
parameters GET,
optionallyRegisteredAs GET;;;

REGISTERED AS {MKMD.mkmMObjectClass 9};**behaviourTemplate MANAGED OBJECT CLASS****DERIVED FROM** template;**CHARACTERIZED BY** behaviourPackage PACKAGE**BEHAVIOUR** behaviourTemplateBehaviour **BEHAVIOUR DEFINED AS !**

A managed object with this behaviour supports the retrieval of the definedAs attribute via the getTextualRepresentation action.

!;;

ATTRIBUTES

definedAs GET;

ACTIONS

getTextualRepresentation;;;

REGISTERED AS {MKMD.mkmMObjectClass 11};**document MANAGED OBJECT CLASS****DERIVED FROM** "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":top;**CHARACTERIZED BY** documentPackage PACKAGE**BEHAVIOUR** documentBehaviour **BEHAVIOUR DEFINED AS !**

When an instance of this class is named subordinate to a system managed object, either the documentName or the documentObjectIdentifier attribute provides unique identification within the scope of that system managed object. When the documentName attribute is used for ITU-T Recommendations or for International Standards, it is formed as specified for a <standard-name> in 8.2(k) of CCITT Rec. X.722 (1992) | ISO/IEC 10165-4:1992.

!;;

ATTRIBUTES

documentName GET,
documentObjectIdentifier GET;;;

CONDITIONAL PACKAGES

documentSpecificationPackage PACKAGE

BEHAVIOUR documentSpecificationBehaviour BEHAVIOUR DEFINED AS !

A managed object with this behaviour supports the retrieval of the specification attribute via the
getTextualRepresentation action.

!;;

ATTRIBUTES

specification GET;

ACTIONS

getTextualRepresentation;

REGISTERED AS {MKMD.mkmPackage 4}; PRESENT IF "document is available as a text string";

REGISTERED AS {MKMD.mkmMObjectClass 14};

managedObjectClassTemplate MANAGED OBJECT CLASS

DERIVED FROM template;

CHARACTERIZED BY managedObjectClassPackage PACKAGE

ATTRIBUTES

derivedFrom GET,
characterizedBy GET,
conditionalPackages GET,
registeredAs GET;;;

REGISTERED AS {MKMD.mkmMObjectClass 5};

nameBindingTemplate MANAGED OBJECT CLASS

DERIVED FROM template;

CHARACTERIZED BY nameBindingPackage PACKAGE

ATTRIBUTES

subordinateObjectClass GET,
namedBySuperiorObjectClass GET,
withAttribute GET,
behaviour GET,
create GET,
delete GET,
registeredAs GET;;;

REGISTERED AS {MKMD.mkmMObjectClass 8};

notificationTemplate MANAGED OBJECT CLASS

DERIVED FROM template;

CHARACTERIZED BY notificationPackage PACKAGE

ATTRIBUTES

behaviour GET,
modeConfirmed GET,
parameters GET,
withInformationSyntax GET,
andAttributeIds GET,
withReplySyntax GET,
registeredAs GET;;;

REGISTERED AS {MKMD.mkmMObjectClass 13};

packageTemplate MANAGED OBJECT CLASS

DERIVED FROM template;

CHARACTERIZED BY packagePackage PACKAGE

ATTRIBUTES

behaviour GET,
attributes GET,
attributeGroups GET,
actions GET,
notifications GET,
optionallyRegisteredAs GET;;;

REGISTERED AS {MKMD.mkmMObjectClass 6};

parameterTemplate MANAGED OBJECT CLASS

DERIVED FROM template;

CHARACTERIZED BY parameterPackage PACKAGE

ATTRIBUTES

context GET,
syntaxOrAttribute GET,
behaviour GET,
optionallyRegisteredAs GET;;;

REGISTERED AS {MKMD.mkmMObjectClass 7};

template MANAGED OBJECT CLASS**DERIVED FROM** "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":top;**CHARACTERIZED BY** templatePackage PACKAGE**BEHAVIOUR** templateBehaviour **BEHAVIOUR DEFINED AS !**

When an instance of this class is named subordinate to a document managed object, the templateName attribute provides unique identification within the scope of that document managed object.

!;;

ATTRIBUTES

templateName GET;;;

CONDITIONAL PACKAGES

templateDefinitionPackage PACKAGE

BEHAVIOUR templateDefinitionPackageBehaviour **BEHAVIOUR DEFINED AS !**

A managed object with this behaviour supports retrieval of the information of the templateDefinition attribute via the getTextualRepresentation action.

!;;

ATTRIBUTES

templateDefinition GET;

ACTIONS

getTextualRepresentation;

REGISTERED AS {MKMD.mkmPackage 3}; **PRESENT IF** !template is represented as text string!;**REGISTERED AS** {MKMD.mkmMObjectClass 3};**A.3.2 Name bindings****aSN1Module-document NAME BINDING****SUBORDINATE OBJECT CLASS** aSN1Module AND SUBCLASSES;**NAMED BY SUPERIOR OBJECT CLASS** document AND SUBCLASSES;**WITH ATTRIBUTE** moduleReference;**REGISTERED AS** {MKMD.mkmNameBinding 4};**document-system NAME BINDING****SUBORDINATE OBJECT CLASS** document AND SUBCLASSES;**NAMED BY SUPERIOR OBJECT CLASS**

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":system AND SUBCLASSES;

WITH ATTRIBUTE documentObjectIdentifier;**REGISTERED AS** {MKMD.mkmNameBinding 6};**document-system2 NAME BINDING****SUBORDINATE OBJECT CLASS** document AND SUBCLASSES;**NAMED BY SUPERIOR OBJECT CLASS**

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":system AND SUBCLASSES;

WITH ATTRIBUTE documentName;**REGISTERED AS** {MKMD.mkmNameBinding 7};**template-document NAME BINDING****SUBORDINATE OBJECT CLASS** template AND SUBCLASSES;**NAMED BY SUPERIOR OBJECT CLASS** document AND SUBCLASSES;**WITH ATTRIBUTE** templateName;**REGISTERED AS** {MKMD.mkmNameBinding 3};**A.3.3 Attributes****actions ATTRIBUTE****WITH ATTRIBUTE SYNTAX** DefinitionASN1Module.Actions;**MATCHES FOR** EQUALITY;**BEHAVIOUR** actionsBehaviour **BEHAVIOUR DEFINED AS !**

This set-valued attribute identifies the actions, and associated parameters, of a package.

!;;

REGISTERED AS {MKMD.mkmAttribute 18};**andAttributeIds ATTRIBUTE****WITH ATTRIBUTE SYNTAX** DefinitionASN1Module.AndAttributeIds;**MATCHES FOR** EQUALITY;**BEHAVIOUR** andAttributeIdsBehaviour **BEHAVIOUR DEFINED AS !**

This set-valued attribute identifies the assignment of attributes to fields of a notification's argument.

!;;

REGISTERED AS {MKMD.mkmAttribute 36};

asn1ModuleContents ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.ASN1ModuleContents;
MATCHES FOR EQUALITY, SUBSTRINGS;
BEHAVIOUR asn1ModuleContentsBehaviour BEHAVIOUR DEFINED AS !

This attribute holds the ASN.1 module contents in textual form.

!;;

REGISTERED AS {MKMD.mkmAttribute 9};

asn1Version ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.ASN1Version;
MATCHES FOR EQUALITY, SUBSTRINGS;
BEHAVIOUR asn1VersionBehaviour BEHAVIOUR DEFINED AS !

This attribute holds a bit string that has one bit set for each ASN.1 version to which the module conforms.

!;;

REGISTERED AS {MKMD.mkmAttribute 45};

attributeGroups ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.AttributeGroups;
MATCHES FOR EQUALITY;
BEHAVIOUR attributeGroupsBehaviour BEHAVIOUR DEFINED AS !

This set-valued attribute identifies the attribute groups, and associated attributes, of a package.

!;;

REGISTERED AS {MKMD.mkmAttribute 17};

attributes ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.Attributes;
MATCHES FOR EQUALITY;
BEHAVIOUR attributesBehaviour BEHAVIOUR DEFINED AS !

This set-valued attribute identifies the attributes, and associated qualifiers and parameters, of a package.

!;;

REGISTERED AS {MKMD.mkmAttribute 16};

behaviour ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.Behaviour;
MATCHES FOR EQUALITY;
BEHAVIOUR behaviourBehaviour BEHAVIOUR DEFINED AS !

This set-valued attribute identifies the behaviours associated with an element of management information.

!;;

REGISTERED AS {MKMD.mkmAttribute 15};

characterizedBy ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.CharacterizedBy;
MATCHES FOR EQUALITY;
BEHAVIOUR characterizedByBehaviour BEHAVIOUR DEFINED AS !

This set-valued attribute identifies the mandatory packages of a managed object class.

!;;

REGISTERED AS {MKMD.mkmAttribute 11};

conditionalPackages ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.ConditionalPackages;
MATCHES FOR EQUALITY;
BEHAVIOUR conditionalPackagesBehaviour BEHAVIOUR DEFINED AS !

This set-valued attribute identifies the conditional packages, and their conditions, of a managed object class.

!;;

REGISTERED AS {MKMD.mkmAttribute 12};

context ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.Context;
MATCHES FOR EQUALITY;
BEHAVIOUR contextBehaviour BEHAVIOUR DEFINED AS !

This attribute identifies the context of a parameter.

!;;

REGISTERED AS {MKMD.mkmAttribute 20};

create ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.Create;
MATCHES FOR EQUALITY;
BEHAVIOUR createBehaviour BEHAVIOUR DEFINED AS !

This attribute identifies the create element specification of a name binding. If no create element is present, then this attribute's value shall be null.

!;;

REGISTERED AS {MKMD.mkmAttribute 25};

definedAs ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.DefinedAs;
MATCHES FOR EQUALITY;
BEHAVIOUR definedAsBehaviour **BEHAVIOUR DEFINED AS !**
 This attribute contains the definition of a behaviour.
 !;;

REGISTERED AS {MKMD.mkmAttribute 32};

delete ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.Delete;
MATCHES FOR EQUALITY;
BEHAVIOUR deleteBehaviour **BEHAVIOUR DEFINED AS !**
 This attribute identifies the delete element specification of a name binding. If no delete element is present, then this attribute's value shall be null.
 !;;

REGISTERED AS {MKMD.mkmAttribute 26};

derivedFrom ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.DerivedFrom;
MATCHES FOR EQUALITY;
BEHAVIOUR derivedFromBehaviour **BEHAVIOUR DEFINED AS !**
 This set-valued attribute identifies the immediate superclasses of a managed object class.
 !;;

REGISTERED AS {MKMD.mkmAttribute 10};

derivedOrWithSyntaxChoice ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.DerivedOrWithSyntaxChoice;
MATCHES FOR EQUALITY;
BEHAVIOUR derivedOrWithSyntaxChoiceBehaviour **BEHAVIOUR DEFINED AS !**
 This attribute identifies the syntax specification of an attribute.
 !;;

REGISTERED AS {MKMD.mkmAttribute 27};

description ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.Description;
MATCHES FOR EQUALITY;
BEHAVIOUR descriptionBehaviour **BEHAVIOUR DEFINED AS !**
 This attribute contains the description of an attribute group.
 !;;

REGISTERED AS {MKMD.mkmAttribute 31};

documentName ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.DocumentName;
MATCHES FOR EQUALITY, SUBSTRINGS;
BEHAVIOUR documentNameBehaviour **BEHAVIOUR DEFINED AS !**
 This attribute identifies with a graphic string the document represented by the managed object. E.g. "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992" identifies the Definition of management information.
 !;;

REGISTERED AS {MKMD.mkmAttribute 38};

documentObjectIdentifier ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.DocumentObjectIdentifier;
MATCHES FOR EQUALITY;
BEHAVIOUR documentObjectIdentifierBehaviour **BEHAVIOUR DEFINED AS !**

This attribute identifies with an ASN.1 object identifier value the document represented by the managed object. E.g. {2 9 3 2} identifies the Definition of management information, CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992 (as specified in the registration tables for Recommendation | International Standard documents for OSI Systems Management published in CCITT Rec. X.722 | ISO/IEC 10165-4).

!;;

REGISTERED AS {MKMD.mkmAttribute 39};

fixed ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.Fixed;
MATCHES FOR EQUALITY;
BEHAVIOUR fixedBehaviour **BEHAVIOUR DEFINED AS !**
 This attribute identifies an attribute group as being fixed or dynamic.
 !;;

REGISTERED AS {MKMD.mkmAttribute 30};

groupElements ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.GroupElements;
MATCHES FOR EQUALITY;
BEHAVIOUR groupElementsBehaviour BEHAVIOUR DEFINED AS !
This set-valued attribute identifies the elements of an attribute group.
!;;

REGISTERED AS {MKMD.mkmAttribute 29};

matchesFor ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.MatchesFor;
MATCHES FOR EQUALITY;
BEHAVIOUR matchesForBehaviour BEHAVIOUR DEFINED AS !
This attribute identifies the matching characteristics of an attribute.
!;;

REGISTERED AS {MKMD.mkmAttribute 28};

modeConfirmed ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.ModeConfirmed;
MATCHES FOR EQUALITY;
BEHAVIOUR modeConfirmedBehaviour BEHAVIOUR DEFINED AS !
This attribute identifies an action or notification as being confirmed or not.
!;;

REGISTERED AS {MKMD.mkmAttribute 33};

moduleReference ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.Identifier;
MATCHES FOR EQUALITY;
BEHAVIOUR moduleReferenceBehaviour BEHAVIOUR DEFINED AS !
This attribute identifies the name of the ASN.1 module.
!;;

REGISTERED AS {MKMD.mkmAttribute 8};

namedBySuperiorObjectClass ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.NamingObjectClass;
MATCHES FOR EQUALITY;
BEHAVIOUR namedBySuperiorObjectClassBehaviour BEHAVIOUR DEFINED AS !
This attribute identifies the superior object classes of a name binding.
!;;

REGISTERED AS {MKMD.mkmAttribute 23};

notifications ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.Notifications;
MATCHES FOR EQUALITY;
BEHAVIOUR notificationsBehaviour BEHAVIOUR DEFINED AS !
This set-valued attribute identifies the notifications, and associated parameters, of a package.
!;;

REGISTERED AS {MKMD.mkmAttribute 19};

optionallyRegisteredAs ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.OptionallyRegisteredAs;
MATCHES FOR EQUALITY;
BEHAVIOUR optionallyRegisteredAsBehaviour BEHAVIOUR DEFINED AS !
This attribute identifies the identifier of the represented element of management information. If no identifier has been assigned, this attribute shall have a null value.
!;;

REGISTERED AS {MKMD.mkmAttribute 43};

parameters ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.Parameters;
MATCHES FOR EQUALITY;
BEHAVIOUR parametersBehaviour BEHAVIOUR DEFINED AS !
This set-valued attribute identifies the parameters associated with an element of management information.
!;;

REGISTERED AS {MKMD.mkmAttribute 13};

registeredAs ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.RegisteredAs;
MATCHES FOR EQUALITY;
BEHAVIOUR registeredAsBehaviour BEHAVIOUR DEFINED AS !
This attribute identifies the identifier of the represented element of management information.
!;;

REGISTERED AS {MKMD.mkmAttribute 14};

specification ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.Specification;

BEHAVIOUR specificationBehaviour **BEHAVIOUR DEFINED AS !**

This attribute contains a representation of only the GDMO and ASN.1 specifications and embedded comments contained within the document.

!;;

REGISTERED AS {MKMD.mkmAttribute 40};

subordinateObjectClass ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.NamingObjectClass;

MATCHES FOR EQUALITY;

BEHAVIOUR subordinateObjectClassBehaviour **BEHAVIOUR DEFINED AS !**

This attribute identifies the subordinate managed object classes of a name binding.

!;;

REGISTERED AS {MKMD.mkmAttribute 22};

syntaxOrAttribute ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.SyntaxOrAttribute;

MATCHES FOR EQUALITY;

BEHAVIOUR syntaxOrAttributeBehaviour **BEHAVIOUR DEFINED AS !**

This attribute identifies the syntax specification of a parameter.

!;;

REGISTERED AS {MKMD.mkmAttribute 21};

templateDefinition ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.TemplateDefinition;

MATCHES FOR EQUALITY;

BEHAVIOUR templateDefinitionBehaviour **BEHAVIOUR DEFINED AS !**

This attribute contains a text string representation of the GDMO template definition. In-line templates are not included in this text string, even if they appear in line in the document represented by the document managed object under which the template managed object is named. It is not necessary that GDMO comments be included in this representation.

!;;

REGISTERED AS {MKMD.mkmAttribute 37};

templateName ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.TemplateName;

MATCHES FOR EQUALITY;

BEHAVIOUR templateNameBehaviour **BEHAVIOUR DEFINED AS !**

This attribute names template managed objects. Its value is that of the label of the GDMO template which is represented.

!;;

REGISTERED AS {MKMD.mkmAttribute 7};

withAttribute ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.WithAttribute;

MATCHES FOR EQUALITY;

BEHAVIOUR withAttributeBehaviour **BEHAVIOUR DEFINED AS !**

This attribute identifies the naming attribute of a name binding.

!;;

REGISTERED AS {MKMD.mkmAttribute 24};

withInformationSyntax ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.WithSyntax;

MATCHES FOR EQUALITY;

BEHAVIOUR withInformationSyntaxBehaviour **BEHAVIOUR DEFINED AS !**

This attribute identifies the syntax of the argument of an action or notification.

!;;

REGISTERED AS {MKMD.mkmAttribute 34};

withReplySyntax ATTRIBUTE

WITH ATTRIBUTE SYNTAX DefinitionASN1Module.WithSyntax;

MATCHES FOR EQUALITY;

BEHAVIOUR withReplySyntaxBehaviour **BEHAVIOUR DEFINED AS !**

This attribute identifies the syntax of the reply to an action or notification.

!;;

REGISTERED AS {MKMD.mkmAttribute 35};

A.3.4 Actions and parameters

getTextualRepresentation ACTION

BEHAVIOUR getTextualRepresentationBehaviour BEHAVIOUR DEFINED AS !

The action returns a textual representation of the information in a single attribute of the managed object to which it is directed. The action response, which may be spread across multiple Linked Replies, consists of a sequence of line images. If there are multiple Linked Replies, the sequences from all of them are logically combined into a single, longer sequence of line images.

For an attribute with the syntax DefinitionASN1Module.TextualRepresentation, the result of retrieving the attribute via this action and, if necessary, combining the contents of multiple Linked Replies, is identical to result of the get attribute value management operation.

!;;

PARAMETERS attributeNotAvailableViaThisAction;

WITH INFORMATION SYNTAX DefinitionASN1Module.AttributeId;

WITH REPLY SYNTAX DefinitionASN1Module.TextualRepresentation;

REGISTERED AS {MKMD.mkmAction 2};

attributeNotAvailableViaThisAction PARAMETER

CONTEXT SPECIFIC-ERROR;

WITH SYNTAX DefinitionASN1Module.Null;

BEHAVIOUR attributeNotAvailableViaThisActionBehaviour BEHAVIOUR DEFINED AS !

This error indicates that retrieval of the requested attribute via the getTextualRepresentation action is not supported.

!;;

REGISTERED AS {MKMD.mkmParameter 1};

A.3.5 Abstract syntax

-- %PRAGMA version BIT STRING {v1990(0), v1994(1)} ::= {v1990, v1994}

DefinitionASN1Module {joint-iso-ccitt(2) ms(9) function(2) part16(16) asn1Modules(2) 1}

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- EXPORTS everything;

IMPORTS

ManagementExtension

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

-- CCITT Rec. X.721 | ISO/IEC 10165-2

;

Actions ::= SET OF SEQUENCE {
 action [0] TemplateLabel,
 parameter [1] TemplateList OPTIONAL}

AdditionalProperties ::= SET OF ManagementExtension

AndAttributeIds ::= SET OF SEQUENCE {
 fieldNameIdentifier,
 attribute TemplateLabel}

ASN1ModuleContents ::= TextualRepresentation

ASN1Version ::= BIT STRING {v1990(0), v1994(1)}

AttributeGroups ::= SET OF SEQUENCE {
 group TemplateLabel,
 attributes TemplateList OPTIONAL}

AttributeId ::= OBJECT IDENTIFIER

Attributes ::= SET OF SEQUENCE {
 attributeLabel TemplateLabel,
 property PropertyList,
 parameterLabel TemplateList OPTIONAL}

Behaviour ::= TemplateList

CharacterizedBy ::= TemplateList

ConditionalPackages ::= SET OF SEQUENCE { label TemplateLabel,
 condition GraphicString}

Context ::= CHOICE {
 keyword SEQUENCE { type-reference DefinedType,
 field Identifier},
 representation ENUMERATED { aCTION-INFO(0),
 aCTION-REPLY(1),
 eVENT-INFO(2),
 eVENT-REPLY(3),
 sPECIFIC-ERROR(4)}}

Create ::= CHOICE {present CreateSpecification, absent NULL}
CreateSpecification ::= SEQUENCE {
createModifier BIT STRING{ withReferenceObject(0),
withAutomaticInstanceNaming(1)}
OPTIONAL,
parameter TemplateList OPTIONAL}
DefinedAs ::= TextualRepresentation
DefinedType ::= SEQUENCE { module [0] Identifier OPTIONAL,
reference [1] Identifier}
DefinedValue ::= SEQUENCE { module [0] Identifier OPTIONAL,
reference [1] Identifier}
Delete ::= CHOICE {present DeleteSpecification, absent NULL}
DeleteSpecification ::= SEQUENCE {
deleteModifier ENUMERATED{ onlyIfNoContainedObjects(0),
deletesContainedObjects(1)} OPTIONAL,
parameter TemplateList OPTIONAL}
DerivedFrom ::= TemplateList
DerivedOrWithSyntaxChoice ::= CHOICE {
derivedFrom [0] TemplateLabel,
withSyntaxChoice [1] DefinedType}
Description ::= TextualRepresentation
DocumentName ::= GraphicString
DocumentObjectIdentifier ::= OptionallyRegisteredAs
Fixed ::= BOOLEAN
GroupElements ::= TemplateList
Identifier ::= PrintableString(FROM("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9" |
"A"|"B"|"C"|"D"|"E"|"F"|"G"|"H"|"I"|"J"|"K"|"L"|"M" |
"N"|"O"|"P"|"Q"|"R"|"S"|"T"|"U"|"V"|"W"|"X"|"Y"|"Z" |
"a"|"b"|"c"|"d"|"e"|"f"|"g"|"h"|"i"|"j"|"k"|"l"|"m" |
"n"|"o"|"p"|"q"|"r"|"s"|"t"|"u"|"v"|"w"|"x"|"y"|"z"|"-""))
LineImage ::= GraphicString -- each value of this type represents one complete line of text
MatchesFor ::= BIT STRING{ equality(0),
ordering(1),
substrings(2),
setComparison(3),
setIntersection(4)}
ModeConfirmed ::= BOOLEAN
NamingObjectClass ::= SEQUENCE{class TemplateLabel,
andSubclasses BOOLEAN DEFAULT FALSE}
Notifications ::= SET OF SEQUENCE {
notification [0] TemplateLabel,
parameter [1] TemplateList OPTIONAL}
Null ::= NULL
OptionallyRegisteredAs ::= CHOICE {present OBJECT IDENTIFIER, absent NULL}
Parameters ::= TemplateList
PropertyList ::= SEQUENCE {
replaceWithDefault [0] BOOLEAN DEFAULT FALSE,
defaultValue [1] EXPLICIT ValueSpecifier OPTIONAL,
initialValue [2] EXPLICIT ValueSpecifier OPTIONAL,
permittedValues [3] DefinedType OPTIONAL,
requiredValues [4] DefinedType OPTIONAL,
getReplace [5] ENUMERATED{
gET(0),
rEPLACE(1),
gET-REPLACE(2)} OPTIONAL,
addRemove [6] ENUMERATED{
aDD(0),
rEMOVE(1),
aDD-REMOVE(2)} OPTIONAL,
setByCreate [7] BOOLEAN DEFAULT FALSE,
additionalProperties [8] AdditionalProperties OPTIONAL}
RegisteredAs ::= OBJECT IDENTIFIER
Specification ::= TextualRepresentation
SyntaxOrAttribute ::= CHOICE {withSyntax [0] DefinedType,
attribute [1] TemplateLabel}
TemplateDefinition ::= TextualRepresentation
TemplateList ::= SET OF TemplateLabel

```

TemplateLabel ::= SEQUENCE{
    localName      TemplateName,
    documentId     CHOICE {
        documentName      DocumentName,
        documentObjectIdentifier DocumentObjectIdentifier
    } OPTIONAL -- align with GDMO document identifier
}
TemplateName ::= PrintableString(FROM("0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" |
    "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" |
    "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z" |
    "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" |
    "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z" | "-" | "/"))
TextualRepresentation ::= SEQUENCE OF LineImage
ValueSpecifier ::= CHOICE{ valueReference [0] DefinedValue,
    derivationRule [1] TemplateLabel}
WithAttribute ::= TemplateLabel
WithSyntax ::= CHOICE { present DefinedType, absent NULL}
END

```

A.4 Discovery managed object definitions

A.4.1 Managed object classes

discoveryObject MANAGED OBJECT CLASS

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

CHARACTERIZED BY discoveryObjectPackage PACKAGE

BEHAVIOUR discoveryObjectBehaviour BEHAVIOUR DEFINED AS !

A managed object with this behaviour responds to mITSearch action requests concerning subtrees of managed objects within a managed system's naming tree. The query specifies a base object and the required scope of the search.

The response is a set of rooted tree structures that together cover the requested scope under the requested base object. Each rooted tree has the distinguished name of its root but does not repeat common name components within its structure, i.e. only the RDN is used to identify subordinate managed objects.

!;;

ATTRIBUTES

discoveryId;

ACTIONS

mITSearch;;;

REGISTERED AS {MKMD.mkmMObjectClass 15};

A.4.2 Name bindings

discovery-system NAME BINDING

SUBORDINATE OBJECT CLASS discoveryObject AND SUBCLASSES;

NAMED BY SUPERIOR OBJECT CLASS

"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992":system AND SUBCLASSES;

WITH ATTRIBUTE discoveryId;

BEHAVIOUR discovery-systemBehaviour BEHAVIOUR DEFINED AS !

The use of a fixed relative distinguished name for a common way to access a discovery managed object is provided by this name binding when the value "DSC1" is used for the repertoire Id attribute.

Subordinate to a given system managed object, there shall be only one allowed value for the discovery Id attribute, namely "DSC1".

!;;

REGISTERED AS {MKMD.mkmNameBinding 5};

A.4.3 Attributes

discoveryId ATTRIBUTE

WITH ATTRIBUTE SYNTAX DiscoveryASN1Module.DiscoveryId;

MATCHES FOR EQUALITY;

BEHAVIOUR discoveryIdBehaviour BEHAVIOUR DEFINED AS !

This attribute is used to identify an instance of the discovery managed object.

!;;

REGISTERED AS {MKMD.mkmAttribute 41};

A.4.4 Actions

mITSearch ACTION

BEHAVIOUR mitSearchBehaviour **BEHAVIOUR DEFINED AS !**

The argument has the base object, the required scope of the search and an optional class request parameter. The response has a tree structure, under any specified base managed object, without repeating common name components (i.e. only RDN components are sent).

It shall, if specified in the request, also report the managed object class type associated with each node in the tree.

!;;

WITH INFORMATION SYNTAX DiscoveryASN1Module.MITSearch;

WITH REPLY SYNTAX DiscoveryASN1Module.MITSearchResponse;

REGISTERED AS {MKMD.mkmAction 1};

A.4.5 Abstract syntax

--%PRAGMA version BIT STRING {v1990(0), v1994(1)} ::= {v1990, v1994}

DiscoveryASN1Module {joint-iso-ccitt(2) ms(9) function(2) part16(16) asn1Modules(2) 2}

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- EXPORTS everything;

IMPORTS

RelativeDistinguishedName

FROM InformationFramework {joint-iso-ccitt(2) ds(5) modules(1) informationFramework(1)}

-- ITU-T Rec. X.501 | ISO/IEC 9594-2

GdmoObjectClass

FROM RepertoireASN1Module {joint-iso-ccitt(2) ms(9) function(2) part16(16) asn1Modules(2) 0}

-- this Recommendation | International Standard

ObjectInstance, Scope

FROM CMIP-1 {joint-iso-ccitt(2) ms(9) cmip(1) modules(0) protocol(3)}

-- ITU-T Rec. X.711 and ISO/IEC 9596-1

;

DiscoveryId ::= GraphicString

MITSearch ::= SEQUENCE {

base ObjectInstance,

discoveryScopeScope,

classRequest **BOOLEAN DEFAULT FALSE**

-- if true, include class with MITSearchResponse

}

MITSearchResponse ::= SET OF RootedNameTree

NameTree ::= SEQUENCE {

rdnInfo RDNInfo,

subordinates SET OF NameTree OPTIONAL}

RDNInfo ::= SEQUENCE {

rdn RelativeDistinguishedName,

moClass GdmoObjectClass OPTIONAL}

RootedNameTree ::= SEQUENCE {

rootObject ObjectInstance,

classOfRoot GdmoObjectClass OPTIONAL,

subordinates SET OF NameTree OPTIONAL}

END

Annexe B

Objets d'Annuaire de connaissance de gestion

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

B.1 Répertoire Directory object definitions

```

--%PRAGMA version BIT STRING {v1990(0), v1994(1)} ::= v1994
RepertoireDirectoryASN1Module {joint-iso-ccitt(2) ms(9) function(2) part16(16) asn1Modules(2) 3}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
-- EXPORTS everything;
IMPORTS
ATTRIBUTE, OBJECT-CLASS CONTENT-RULE
    FROM InformationFramework {joint-iso-ccitt(2) ds(5) modules(1) informationFramework(1) 2}
    -- ITU-T Rec. X.501 | ISO/IEC 9594-2
applicationEntity
    FROM SelectedObjectClasses {joint-iso-ccitt(2) ds(5) modules(1) selectedObjectClasses(6) 2}
    -- ITU-T Rec. X.521 | ISO/IEC 9594-7
bitStringMatch
    FROM SelectedAttributeTypes {joint-iso-ccitt(2) ds(5) modules(1) selectedAttributeTypes(5) 2}
    -- ITU-T Rec. X.520 | ISO/IEC 9594-6
ObjectInstance
    FROM CMIP-1 {joint-iso-ccitt(2) ms(9) cmip(1) modules(0) protocols(3)}
    -- CCITT Rec. X.711 and ISO/IEC 9596-1
FunctionalUnits, ProtocolVersion
    FROM CMIP-A-ASSOCIATE-Information {joint-iso-ccitt(2) ms(9) cmip(1) modules (0)
    AssociateUserInfo(1)} -- CCITT Rec. X.711 and ISO/IEC 9596-1
FunctionalUnitPackage
    FROM SMASE-A-ASSOCIATE-Information {joint-iso-ccitt(2) ms(9) smo(0) negotiationAbstractSyntax(1)
    version1(1)} -- CCITT Rec. X.701 | ISO/IEC 10040
mkmDirectoryObjectClass, mkmDirectoryAttributeType
    FROM MKMD {joint-iso-ccitt(2) ms(9) function(2) part16(16) asn1Modules(2) 5}
    -- this Recommendation | International Standard
GdmoObjectClass
    FROM RepertoireASN1Module {joint-iso-ccitt(2) ms(9) function(2) part16(16) asn1Modules(2) 0}
    -- this Recommendation | International Standard
;
cMISE OBJECT-CLASS ::= {
    KIND auxiliary
    MUST CONTAIN { supportedCmipVersion |
    supportedCmipProfiles |
    supportedCmisFunctionalUnits}
    ID { mkmDirectoryObjectClass 1}
sMASE OBJECT-CLASS ::= {
    KIND auxiliary
    MUST CONTAIN { supportedSmaseFunctionalUnits}
    MAY CONTAIN { mitMoList |
    supportsMKMglobalNames}
    ID { mkmDirectoryObjectClass 2}
-- The "cMISE" and "sMASE" Directory auxiliary object
-- classes should be used together with the Directory
-- structural object class "applicationEntity".
InstanceIdList ::= SET OF ObjectInstance
managementServiceElementsCR CONTENT-RULE ::= {
    STRUCTURAL OBJECT CLASS applicationEntity
    AUXILIARY OBJECT CLASS { sMASE | cMISE}}
mitMoList ATTRIBUTE ::= {
    WITH SYNTAX MitMo
    ID { mkmDirectoryAttributeType 1}
MitMo ::= SEQUENCE {
    objectClass GdmoObjectClass,
    instances InstanceIdList OPTIONAL}

```

```

supportedCmipProfiles ATTRIBUTE ::= {
    WITH SYNTAX BIT STRING
    EQUALITY MATCHING RULE bitStringMatch
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 43}
supportedCmipVersion ATTRIBUTE ::= {
    WITH SYNTAX ProtocolVersion
    EQUALITY MATCHING RULE bitStringMatch
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 2}}
supportedCmisFunctionalUnits ATTRIBUTE ::= {
    WITH SYNTAX FunctionalUnits
    EQUALITY MATCHING RULE bitStringMatch
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 3}}
supportedSmaseFunctionalUnits ATTRIBUTE ::= {
    WITH SYNTAX FunctionalUnitPackage
    ID { mkmDirectoryAttributeType 5}}
supportsMKMglobalNames ATTRIBUTE ::= {
    WITH SYNTAX BOOLEAN
    EQUALITY MATCHING RULE booleanMatch
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 44}}
END

```

B.2 Definition Directory object definitions

```

--%PRAGMA version BIT STRING {v1990(0), v1994(1)} ::= v1994
DefinitionDirectoryASN1Module {joint-iso-ccitt(2) ms(9) function(2) part16(16) asn1Modules(2) 4}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
-- EXPORTS everything;
IMPORTS
ATTRIBUTE, OBJECT-CLASS, CONTENT-RULE, NAME-FORM, objectIdentifierMatch
    FROM InformationFramework {joint-iso-ccitt(2) ds(5) modules(1) informationFramework(1) 2}
    -- ITU-T Rec. X.501 | ISO/IEC 9594-2
bitStringMatch, booleanMatch, caseExactMatch, commonName, directoryStringFirstComponentMatch, integerMatch
    FROM SelectedAttributeTypes {joint-iso-ccitt(2) ds(5) modules(1) selectedAttributeTypes(5) 2}
    -- ITU-T Rec. X.520 | ISO/IEC 9594-6
    -- Because directoryStringFirstComponentMatch is a case-ignore matching rule, some matches
    -- found by its use could fail an exact-case matching test.
Actions, AndAttributeIds, Attributes, AttributeGroups, ConditionalPackages, Context, Create, Delete,
DerivedOrWithSyntaxChoice, Identifier, NamingObjectClass, Notifications, OptionallyRegisteredAs, RegisteredAs,
SyntaxOrAttribute, TemplateLabel, TextualRepresentation, WithSyntax
    FROM DefinitionASN1Module {joint-iso-ccitt(2) ms(9) function(2) part16(16) asn1Modules(2) 1}
    -- this Recommendation | International Standard
mkmDirectoryObjectClass, mkmDirectoryAttributeType, mkmDirectoryNameForm
    FROM MKMD {joint-iso-ccitt(2) ms(9) function(2) part16(16) asn1Modules(2) 5}
    -- this Recommendation | International Standard
;
-- Definition of the required Directory object classes
asn1Module OBJECT-CLASS ::= {
    KIND auxiliary
    MUST CONTAIN { moduleReference |
        asn1ModuleContents |
        asn1Version |
        optionallyRegisteredAs}
    ID { mkmDirectoryObjectClass 15}}
managementAction OBJECT-CLASS ::= {
    SUBCLASS OF managementTemplate
    KIND auxiliary
    MUST CONTAIN { behaviour |
        modeConfirmed |
        parameters |
        withInformationSyntax |
        withReplySyntax |
        registeredAs}
    ID { mkmDirectoryObjectClass 13}}

```

```

managementAttributeGroup OBJECT-CLASS ::= {
    SUBCLASS OF    managementTemplate
    KIND           auxiliary
    MUST CONTAIN   { groupElements |
                    fixed |
                    description |
                    registeredAs}
    ID             { mkmDirectoryObjectClass 11}}
managementAttribute OBJECT-CLASS ::= {
    SUBCLASS OF    managementTemplate
    KIND           auxiliary
    MUST CONTAIN   { derivedOrWithSyntaxChoice |
                    matchesFor |
                    behaviour |
                    parameters |
                    optionallyRegisteredAs}
    ID             { mkmDirectoryObjectClass 10}}
managementBehaviour OBJECT-CLASS ::= {
    SUBCLASS OF    managementTemplate
    KIND           auxiliary
    MUST CONTAIN   { definedAs}
    ID             { mkmDirectoryObjectClass 12}}
managementDocument OBJECT-CLASS ::= {
    KIND           auxiliary
    MUST CONTAIN   { documentName |
                    documentObjectIdentifier}
    MAY CONTAIN   { specification}
    ID             { mkmDirectoryObjectClass 4}}
managementNameBinding OBJECT-CLASS ::= {
    SUBCLASS OF    managementTemplate
    KIND           auxiliary
    MUST CONTAIN   { subordinateObjectClass |
                    namedBySuperiorObjectClass |
                    withAttribute |
                    behaviour |
                    create |
                    delete |
                    registeredAs}
    ID             { mkmDirectoryObjectClass 9}}
managementNotification OBJECT-CLASS ::= {
    SUBCLASS OF    managementTemplate
    KIND           auxiliary
    MUST CONTAIN   { behaviour |
                    modeConfirmed |
                    parameters |
                    withInformationSyntax |
                    andAttributeIds |
                    withReplySyntax |
                    registeredAs}
    ID             { mkmDirectoryObjectClass 14}}
managementObjectClass OBJECT-CLASS ::= {
    SUBCLASS OF    managementTemplate
    KIND           auxiliary
    MUST CONTAIN   { derivedFrom |
                    characterizedBy |
                    conditionalPackages |
                    registeredAs}
    ID             { mkmDirectoryObjectClass 6}}
managementPackage OBJECT-CLASS ::= {
    SUBCLASS OF    managementTemplate
    KIND           auxiliary
    MUST CONTAIN   { behaviour |
                    attributes |
                    attributeGroups |
                    actions |
                    notifications |
                    optionallyRegisteredAs}
    ID             { mkmDirectoryObjectClass 7}}

```

```

managementParameter OBJECT-CLASS ::= {
    SUBCLASS OF    managementTemplate
    KIND           auxiliary
    MUST CONTAIN   { context |
                    syntaxOrAttribute |
                    behaviour |
                    optionallyRegisteredAs}
    ID             { mkmDirectoryObjectClass 8}}
managementTemplate OBJECT-CLASS ::= {
    KIND           auxiliary
    MUST CONTAIN   { templateName}
    MAY CONTAIN    { templateDefinition}
    ID             { mkmDirectoryObjectClass 5}}
registeredInformation OBJECT-CLASS ::= {
    SUBCLASS OF    top
    MUST CONTAIN   { commonName}
    MAY CONTAIN    { nameForm |
                    informationStatus |
                    additionalInformation}
    ID             { mkmDirectoryObjectClass 3}}
-- Example structure rules for the registeredInformation Directory object class
-- using this name form are provided in Annex H
registeredInformationNameForm NAME-FORM ::= {
    NAMES          registeredInformation
    WITH ATTRIBUTES { commonName}
    ID             { mkmDirectoryNameForm 1}}
-- Definition of DIT content rules
-- Only one of the specified auxiliary object classes can be
-- included in a given entry of class registeredInformation
registeredManagementInformationCR CONTENT-RULE ::= {
    STRUCTURAL OBJECT CLASS    registeredInformation
    AUXILIARY OBJECT CLASS     { managementDocument |
                                managementTemplate |
                                managementObjectClass |
                                managementPackage |
                                managementParameter |
                                managementNameBinding |
                                managementAttribute |
                                managementAttributeGroup |
                                managementBehaviour |
                                managementAction |
                                managementNotification |
                                asn1Module}}
-- Definition of used attributes
actions ATTRIBUTE ::= {
    WITH SYNTAX    Actions
    SINGLE VALUE   TRUE
    ID             { mkmDirectoryAttributeType 6}}
additionalInformation ATTRIBUTE ::= {
    WITH SYNTAX    AdditionalInformation
    SINGLE VALUE   TRUE
    ID             { mkmDirectoryAttributeType 7}}
AdditionalInformation ::= SEQUENCE {
    creationDate    GeneralizedTime OPTIONAL,
    comment         GraphicString OPTIONAL}
andAttributeIds ATTRIBUTE ::= {
    WITH SYNTAX    AndAttributeIds
    SINGLE VALUE   TRUE
    ID             { mkmDirectoryAttributeType 8}}
asn1ModuleContents ATTRIBUTE ::= {
    WITH SYNTAX    TextualRepresentation
    ID             { mkmDirectoryAttributeType 14}}
asn1Version ATTRIBUTE ::= {
    WITH SYNTAX    BIT STRING
    EQUALITY MATCHING RULE    bitStringMatch
    SINGLE VALUE   TRUE
    ID             { mkmDirectoryAttributeType 45}}

```

```

attributeGroups ATTRIBUTE ::= {
    WITH SYNTAX    AttributeGroups
    ID             { mkmDirectoryAttributeType 9}}
attributes ATTRIBUTE ::= {
    WITH SYNTAX    Attributes
    SINGLE VALUE   TRUE
    ID             { mkmDirectoryAttributeType 10}}
behaviour ATTRIBUTE ::= {
    WITH SYNTAX    TemplateLabel
    EQUALITY MATCHING RULE  directoryStringFirstComponentMatch
    ID             { mkmDirectoryAttributeType 11}}
characterizedBy ATTRIBUTE ::= {
    WITH SYNTAX    TemplateLabel
    EQUALITY MATCHING RULE  directoryStringFirstComponentMatch
    ID             { mkmDirectoryAttributeType 12}}
conditionalPackages ATTRIBUTE ::= {
    WITH SYNTAX    ConditionalPackages
    ID             { mkmDirectoryAttributeType 13}}
context ATTRIBUTE ::= {
    WITH SYNTAX    Context
    SINGLE VALUE   TRUE
    ID             { mkmDirectoryAttributeType 15}}
create ATTRIBUTE ::= {
    WITH SYNTAX    Create
    SINGLE VALUE   TRUE
    ID             { mkmDirectoryAttributeType 16}}
definedAs ATTRIBUTE ::= {
    WITH SYNTAX    TextualRepresentation
    ID             { mkmDirectoryAttributeType 17}}
delete ATTRIBUTE ::= {
    WITH SYNTAX    Delete
    SINGLE VALUE   TRUE
    ID             { mkmDirectoryAttributeType 18}}
derivedFrom ATTRIBUTE ::= {
    WITH SYNTAX    TemplateLabel
    EQUALITY MATCHING RULE  directoryStringFirstComponentMatch
    ID             { mkmDirectoryAttributeType 19}}
derivedOrWithSyntaxChoice ATTRIBUTE ::= {
    WITH SYNTAX    DerivedOrWithSyntaxChoice
    SINGLE VALUE   TRUE
    ID             { mkmDirectoryAttributeType 20}}
description ATTRIBUTE ::= {
    WITH SYNTAX    TextualRepresentation
    ID             { mkmDirectoryAttributeType 21}}
documentName ATTRIBUTE ::= {
    WITH SYNTAX    GraphicString
    EQUALITY MATCHING RULE  directoryStringFirstComponentMatch
    SINGLE VALUE   TRUE
    ID             { mkmDirectoryAttributeType 22}}
documentObjectIdentifier ATTRIBUTE ::= {
    WITH SYNTAX    OBJECT IDENTIFIER
    EQUALITY MATCHING RULE  objectIdentifierMatch
    SINGLE VALUE   TRUE
    ID             { mkmDirectoryAttributeType 23}}
fixed ATTRIBUTE ::= {
    WITH SYNTAX    booleanSyntax
    EQUALITY MATCHING RULE  booleanMatch
    SINGLE VALUE   TRUE
    ID             { mkmDirectoryAttributeType 24}}
groupElements ATTRIBUTE ::= {
    WITH SYNTAX    TemplateLabel
    EQUALITY MATCHING RULE  directoryStringFirstComponentMatch
    ID             { mkmDirectoryAttributeType 25}}
informationStatus ATTRIBUTE ::= {
    WITH SYNTAX    InformationStatus
    EQUALITY MATCHING RULE  integerMatch
    SINGLE VALUE   TRUE
    ID             { mkmDirectoryAttributeType 34}}

```

```

InformationStatus ::= ENUMERATED {
    active(0), deleted(1), preliminary(2)}
matchesFor ATTRIBUTE ::= {
    WITH SYNTAX MatchesFor
    EQUALITY MATCHING RULE bitStringMatch
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 26}}
modeConfirmed ATTRIBUTE ::= {
    WITH SYNTAX booleanSyntax
    EQUALITY MATCHING RULE booleanMatch
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 27}}
moduleReference ATTRIBUTE ::= {
    WITH SYNTAX Identifier
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 28}}
namedBySuperiorObjectClass ATTRIBUTE ::= {
    WITH SYNTAX NamingObjectClass
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 29}}
nameForm ATTRIBUTE ::= {
    WITH SYNTAX PrintableString
    EQUALITY MATCHING RULE caseExactMatch
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 30}}
notifications ATTRIBUTE ::= {
    WITH SYNTAX Notifications
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 31}}
optionallyRegisteredAs ATTRIBUTE ::= {
    WITH SYNTAX OptionallyRegisteredAs
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 46}}
parameters ATTRIBUTE ::= {
    WITH SYNTAX TemplateLabel
    EQUALITY MATCHING RULE directoryStringFirstComponentMatch
    ID { mkmDirectoryAttributeType 32}}
registeredAs ATTRIBUTE ::= {
    WITH SYNTAX RegisteredAs
    EQUALITY MATCHING RULE objectIdentifierMatch
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 33}}
specification ATTRIBUTE ::= {
    WITH SYNTAX TextualRepresentation
    ID { mkmDirectoryAttributeType 35}}
subordinateObjectClass ATTRIBUTE ::= {
    WITH SYNTAX NamingObjectClass
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 36}}
syntaxOrAttribute ATTRIBUTE ::= {
    WITH SYNTAX SyntaxOrAttribute
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 37}}
templateDefinition ATTRIBUTE ::= {
    WITH SYNTAX TextualRepresentation
    ID { mkmDirectoryAttributeType 38}}
templateName ATTRIBUTE ::= {
    WITH SYNTAX TemplateLabel
    EQUALITY MATCHING RULE directoryStringFirstComponentMatch
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 39}}
withAttribute ATTRIBUTE ::= {
    WITH SYNTAX TemplateLabel
    EQUALITY MATCHING RULE directoryStringFirstComponentMatch
    SINGLE VALUE TRUE
    ID { mkmDirectoryAttributeType 40}}

```

ISO/CEI 10164-16 : 1997 (F)

```
withInformationSyntax ATTRIBUTE ::= {  
    WITH SYNTAX WithSyntax  
    SINGLE VALUE TRUE  
    ID { mkmDirectoryAttributeType 41}  
withReplySyntax ATTRIBUTE ::= {  
    WITH SYNTAX WithSyntax  
    SINGLE VALUE TRUE  
    ID { mkmDirectoryAttributeType 42}  
END
```

Annexe C

Formulaire MCS

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

C.1 Introduction

C.1.1 Purpose and structure

The Management Conformance Summary (MCS) is a statement by a supplier that identifies an implementation and provides information on whether the implementation claims conformance to any of the listed set of documents that specify conformance requirements to OSI management.

The MCS proforma is a document in the form of a questionnaire that when completed by the supplier of an implementation becomes the MCS.

C.1.2 Instructions for completing the MCS proforma to produce an MCS

The supplier of the implementation shall enter an explicit statement in each of the boxes provided. Specific instruction is provided in the text which precedes each table.

C.1.3 Symbols, abbreviations and terms

For all annexes of this Recommendation | International Standard, the following common notations, defined in CCITT Rec. X.291 and ISO/IEC 9646-2 and ITU-T Rec. X.296 and ISO/IEC 9646-7, are used for the Status column:

m	Mandatory
o	Optional
c	Conditional
x	Prohibited
–	Not applicable or out of scope

NOTES

- 'c', 'm' and 'o' are prefixed by "c:" when nested under a conditional or optional item of the same table;
- 'o' may be suffixed by ".N" (where N is a unique number) for selectable options among a set of status values. Support of at least one of the choices (from the items with the same value of N) is required.

For all annexes of this Recommendation | International Standard, the following common notations, defined in CCITT Rec. X.291 and ISO/IEC 9646-2 and ITU-T Rec. X.296 and ISO/IEC 9646-7, are used for the Support column:

Y	Implemented
N	Not implemented
–	No answer required
Ig	The item is ignored (i.e. processed syntactically but not semantically)

C.1.4 Table format

Some of the tables in this Recommendation | International Standard have been split because the information is too wide to fit on the page. Where this occurs, the index numbers of the first block of columns are the index numbers of the corresponding rows of the remaining blocks of columns. A complete table reconstructed from the constituent parts should have the following layout:

Index	First block of columns	Second block of columns	Etc.
-------	------------------------	-------------------------	------

In this Recommendation | International Standard the constituent parts of the table appear consecutively, starting with the first block of columns.

ISO/CEI 10164-16 : 1997 (F)

When a table with subrows is too wide to fit on a page, the continuation table(s) have been constructed with index numbers identical to the index numbers in the corresponding rows of the first table and with subindex numbers corresponding to the subrows within each indexed row. For example, if Table X.1 has 2 rows and the continuation of Table X.1 has 2 subrows for each row, the tables are presented as follows:

Table X.1 – Title

Index	A	B	C	D	Support		
					E	F	G
1	a	b	–				
2	a	b	–				

Table X.1 (continued) – Title

Index	Subindex	H	I	J	K	L
1	1.1	h	i	j		
	1.2	h	i	j		
2	2.1	h	i	j		
	2.1	h	i	j		

A complete table reconstructed from the constituent parts should have the following layout:

Index	A	B	C	D	Support			Subindex	H	I	J	K	L
					E	F	G						
1	a	b	–					1.1	h	I	j		
								1.2	h	I	j		
2	a	b	–					2.1	h	I	j		
								2.2	h	I	j		

References made to cells within tables shall be interpreted as references within reconstructed tables. In the example, above, the reference X.1/1d corresponds with the fourth blank cell (that is, the cell in column G) for row with Index 1 and X.1/1.2b corresponds with the second blank cell (in column L) for row with Subindex 1.2.

C.2 Identification of the implementation

C.2.1 Date of statement

The supplier of the implementation shall enter the date of this statement in the following box. Use the format DD-MM-YYYY.

Date of statement

C.2.2 Identification of the implementation

The supplier of the implementation shall enter information necessary to uniquely identify the implementation and the system(s) in which it may reside, in the following box.

--

C.2.3 Contact

The supplier of the implementation shall provide information on whom to contact if there are any queries concerning the contents of the MCS or any referenced implementation conformance statement, in the following box.

C.3 Identification of the documents in which the management information is defined

The supplier of the implementation shall enter the title, reference number and date of the publication of the documents which specify the management information to which conformance is claimed, in the following box.

ITU-T Recommendation X.750 (1996) | ISO/IEC 10164-16:1996, Management knowledge management function

(Other documents to which conformance is claimed)

C.3.1 Technical corrigenda implemented

The supplier of the implementation shall enter the reference numbers of implemented technical corrigenda which modify the identified documents, in the following box.

C.3.2 Amendments implemented

The supplier of the implementation shall state the titles and reference numbers of implemented amendments to the identified documents, in the following box.

C.4 Management conformance summary

The supplier of the implementation shall state the capabilities and features supported and provide a summary of conformance claims to this Recommendation | International Standard using the tables in this annex.

The supplier of the implementation shall specify the roles that are supported, in Table C.1.

Table C.1 – Roles

Index	Roles supported	Status	Support	Additional information
1	Manager role	o.1		
2	Agent role	o.1		
3	Directory information user role	o.1		
4	Directory information provider role	o.1		

ISO/CEI 10164-16 : 1997 (F)

The supplier of an implementation that claims support for this Recommendation | International Standard in only the Directory information user role or the Directory information provider role or both is not required to specify support in Tables C.2 to C.8.

The supplier of the implementation shall specify support for management information in the manager role, in Table C.2.

Table C.2 – Manager role minimum conformance requirement

Index	Item	Status	Support	Additional information
1	Operations on managed objects	c4		
2	Object creation notification	c4		
3	Object deletion notification	c4		
4	Attribute value change notification	c4		
c4: if C.1/1a then o.2 else –				

The supplier of the implementation shall specify support for management information in the agent role, in Table C.3. If additional subclasses of log records are supported, the supplier of the implementation shall list the classes in the Additional information column.

Table C.3 – Agent role minimum conformance requirement

Index	Item	Status	Support	Additional information
1	managedObjectClassRepertoire managed object class	c5		
2	repertoire managed object class	c6		
3	actionTemplate managed object class	c6		
4	asn1module	c6		
5	attributeGroupTemplate managed object class	c6		
6	attributeTemplate managed object class	c6		
7	behaviourTemplate managed object class	c6		
8	document managed object class	c7		
9	managedObjectClassTemplate managed object class	c6		
10	nameBindingTemplate managed object class	c6		
11	notificationTemplate managed object class	c6		
12	packageTemplate managed object class	c6		
13	parameterTemplate managed object class	c6		
14	Template managed object class	c6		
15	Discovery managed object class	c6		
16	Subclasses of log records associated with notifications emitted by the above managed objects	c8		

c5: if C.3/2a then m else (if C.1/2a then o.3 else –)

NOTE 1 – Condition c5 makes it mandatory to support the managedObjectClassRepertoire managed object class if the repertoire managed object class is supported.

c6: if C.1/2a then o.3 else –

c7: if (C.3/3a or C.3/4a or C.3/5a or C.3/6a or C.3/7a or C.3/9a or C.3/10a or C.3/11a or C.3/12a or C.3/13a or C.3/14a) then m else (if C.1/2a then o.3 else –)

NOTE 2 – Condition c7 makes it mandatory to support the document managed object class if at least one of the definition managed object classes is supported.

c8: if C.1/2a and C.4/1a then m else –

NOTE 3 – Condition c8 makes it mandatory, if logging is supported, to support the event log records associated with the notifications supported.

Table C.4 – Logging of event records

Index		Status	Support	Additional information
1	Does the implementation support logging of event records in agent role?	c9		
c9: if C.1/2a then o else –				

The supplier of the implementation shall provide information on claims of conformance to any of the documents summarized in Tables C.5 to C.8. For each document that the supplier of the implementation claims conformance to, the corresponding conformance statement(s) shall be completed, or referenced by, the MCS. The supplier of the implementation shall complete the Support, Table numbers and Additional information columns.

In Tables C.5 to C.8, the Status column is used to indicate whether the supplier of the implementation is required to complete the referenced tables or referenced items. Conformance requirements are as specified in the referenced tables or referenced items and are not changed by the value of the MCS Status column. The Support column is used by the supplier of the implementation to indicate completion of the referenced tables or referenced items.

Table C.5 – PICS support summary

Index	Identification of the document that includes the PICS proforma	Table numbers of PICS proforma	Description	Constraints and values	Status	Support	Table numbers of PICS	Additional information
1	CCITT Rec. X.730 ISO/IEC 10164-1	Annex E all tables	Systems management application context	OBJECT IDENTIFIER	m			

NOTE – Conformance to the MAPDUs defined in this Recommendation | International Standard can be claimed by completing the corresponding tables in the MICS and MOCS annexes of the referenced Recommendations | International Standards.

Table C.6 – MOCS support summary

Index	Identification of the document that includes the MOCS proforma	Table numbers of MOCS proforma	Description	Constraints and values	Status	Support	Table numbers of MOCS	Additional information
1	CCITT Rec. X.730 ISO/IEC 10164-1	Annex C all tables	objectCreation, objectDeletion and attributeValueChange records	–	c10			
2	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables E.1-E.4	managedObjectClass Repertoire managed object class	–	c11			
3	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables E.5-E.9	repertoire managed object class	–	c12			
4	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables E.10-E.15	actionTemplate managed object class	–	c13			
5	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables E.16-E.20	asn1module	–	c14			
6	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables E.21-E.26	attributeGroupTemplate managed object class	–	c15			
7	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables E.27-E.32	attributeTemplate managed object class	–	c16			

Table C.6 (concluded) – MOCS support summary

Index	Identification of the document that includes the MOCS proforma	Table numbers of MOCS proforma	Description	Constraints and values	Status	Support	Table numbers of MOCS	Additional information
8	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables E.33-E.38	behaviourTemplate managed object class	–	c17			
9	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables E.39-E.44	document managed object class	–	c18			
10	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables E.45-E.50	managedObjectClassTemplate managed object class	–	c19			
11	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables E.51-E.56	nameBindingTemplate managed object class	–	c20			
12	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables E.57-E.62	notificationTemplate managed object class	–	c21			
13	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables E.63-E.68	packageTemplate managed object class	–	c22			
14	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables E.69-E.74	parameterTemplate managed object class	–	c23			
15	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables E.75-E.80	template managed object class	–	c24			
16	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables E.81-E.84	discovery managed object class	–	c25			
c10: if C.3/16a then m else – c11: if C.3/1a then m else – c12: if C.3/2a then m else – c13: if C.3/3a then m else – c14: if C.3/4a then m else – c15: if C.3/5a then m else – c16: if C.3/6a then m else – c17: if C.3/7a then m else – c18: if C.3/8a then m else – c19: if C.3/9a then m else – c20: if C.3/10a then m else – c21: if C.3/11a then m else – c22: if C.3/12a then m else – c23: if C.3/13a then m else – c24: if C.3/14a then m else – c25: if C.3/15a then m else –								

Table C.7 – MRCS support summary

Index	Identification of the document that includes the MRCS proforma	Table numbers of MRCS proforma	Description	Constraints and values	Status	Support	Table numbers of MRCS	Additional information
1	CCITT Rec. X.735 ISO/IEC 10164-6	Item D.1/1	logRecord-log name binding	–	c26			
2	ITU-T Rec. X.750 ISO/IEC 10164-16	Item F.1/1	repertoire-system name binding	–	c27			
3	ITU-T Rec. X.750 ISO/IEC 10164-16	Item F.1/2	managedObjectClassRepertoire-repertoire name binding	–	c28			
4	ITU-T Rec. X.750 ISO/IEC 10164-16	Item F.1/3	template-document name binding	–	c29			
5	ITU-T Rec. X.750 ISO/IEC 10164-16	Item F.1/4	aSN1Module-document name binding	–	c30			
6	ITU-T Rec. X.750 ISO/IEC 10164-16	Item F.1/5	discovery-system name binding	–	c31			
7	ITU-T Rec. X.750 ISO/IEC 10164-16	Item F.1/6	document-system name binding	–	c32			
8	ITU-T Rec. X.750 ISO/IEC 10164-16	Item F.1/7	document-system2 name binding	–	c32			
c26: if C.4/1a then o else – c27: if C.3/2a then o else – c28: if C.3/1a then o else – c29: if (C.3/3a or C.3/5a or C.3/6a or C.3/7a or C.3/9a or C.3/10a or C.3/11a or C.3/12a or C.3/13a or C.3/14a) then o else – c30: if C.3/4a then o else – c31: if C.3/15a then o else – c32: if C.3/8a then o else –								

Table C.8 – MICS support summary

Index	Identification of the document that includes the MICS proforma	Table numbers of MICS proforma	Description	Constraints and values	Status	Support	Table numbers of MICS	Additional information
1	ITU-T Rec. X.750 ISO/IEC 10164-16	Tables D.1 -D.3	Management operations	–	c33			
2	CCITT Rec. X.730 ISO/IEC 10164-1	Table B.1	objectCreation, objectDeletion and attributeValueChange notifications	–	c34			
c33: if C.2/1a then m else – c34: if C.2/2a or C.2/3a or C.2/4a then m else –								

Annexe D

Formulaire MICS

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

D.1 Introduction

The purpose of this MICS proforma is to provide a mechanism for a supplier of an implementation which claims conformance, in the manager role, to management information specified in this Recommendation | International Standard, to provide conformance information in a standard form.

D.2 Instructions for completing the MICS proforma to produce an MICS

The MICS proforma contained in this annex is comprised of information in tabular form, in accordance with ITU-T Rec. X.724 | ISO/IEC 10165-6. In addition to the general guidance given in ITU-T Rec. X.724 | ISO/IEC 10165-6, the Additional information column shall be used to identify the managed object classes for which the management operations are supported. The supplier of the implementation shall state which items are supported in the tables that follow and if necessary, provide additional information.

D.3 Statement of conformance to the management information

D.3.1 Attributes

The specifier of a manager role implementation that claims to support management operations on the attributes specified in this Recommendation | International Standard shall import a copy of Table D.1 and complete it.

Table D.1 – Attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	o		o	
2	nameBinding	{MKMD.dmiAttribute 63}	–	–		o	
3	packages	{MKMD.dmiAttribute 66}	–	o		o	
4	allomorphs	{MKMD.dmiAttribute 50}	–	–		o	
5	implementedValues	{MKMD.mkmAttribute 6}	–	–		o.4	
6	managedObjectClassRepertoireId	{MKMD.mkmAttribute 2}	–	–		o.4	
7	repertoireId	{MKMD.mkmAttribute 1}	–	–		o.4	
8	supportedCmipProfiles	{MKMD.mkmAttribute 4}	–	–		o.4	
9	supportedConditionalPackageList	{MKMD.mkmAttribute 5}	–	–		o.4	
10	supportedManagedObjectClassList	{MKMD.mkmAttribute 3}	–	–		o.4	
11	supportedNameBindingList	{MKMD.mkmAttribute 42}	–	–		o.4	
12	supportedRelationshipClassList	{MKMD.mkmAttribute 44}	–	–		o.4	
13	actions	{MKMD.mkmAttribute 18}	–	–		o.4	
14	andAttributeIds	{MKMD.mkmAttribute 36}	–	–		o.4	
15	asn1ModuleContents	{MKMD.mkmAttribute 9}	–	–		o.4	
16	asn1Version	{MKMD.mkmAttribute 45}	–	–		o.4	
17	attributeGroups	{MKMD.mkmAttribute 17}	–	–		o.4	
18	attributes	{MKMD.mkmAttribute 16}	–	–		o.4	

Table D.1 (continued) – Attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
19	behaviour	{MKMD.mkmAttribute 15}	–	–		o.4	
20	characterizedBy	{MKMD.mkmAttribute 11}	–	–		o.4	
21	conditionalPackages	{MKMD.mkmAttribute 12}	–	–		o.4	
22	context	{MKMD.mkmAttribute 20}	–	–		o.4	
23	create	{MKMD.mkmAttribute 25}	–	–		o.4	
24	definedAs	{MKMD.mkmAttribute 32}	–	–		o.4	
25	delete	{MKMD.mkmAttribute 26}	–	–		o.4	
26	derivedFrom	{MKMD.mkmAttribute 10}	–	–		o.4	
27	derivedOrWithSyntaxChoice	{MKMD.mkmAttribute 27}	–	–		o.4	
28	description	{MKMD.mkmAttribute 31}	–	–		o.4	
29	documentName	{MKMD.mkmAttribute 38}	–	–		o.4	
30	documentObjectIdentifier	{MKMD.mkmAttribute 39}	–	–		o.4	
31	fixed	{MKMD.mkmAttribute 30}	–	–		o.4	
32	groupElements	{MKMD.mkmAttribute 29}	–	–		o.4	
33	matchesFor	{MKMD.mkmAttribute 28}	–	–		o.4	
34	modeConfirmed	{MKMD.mkmAttribute 33}	–	–		o.4	
35	moduleReference	{MKMD.mkmAttribute 8}	–	–		o.4	
36	namedBySuperiorObjectClass	{MKMD.mkmAttribute 23}	–	–		o.4	
37	notifications	{MKMD.mkmAttribute 19}	–	–		o.4	
38	optionallyRegisteredAs	{MKMD.mkmAttribute 43}	–	–		o.4	
39	parameters	{MKMD.mkmAttribute 13}	–	–		o.4	
40	registeredAs	{MKMD.mkmAttribute 14}	–	–		o.4	
41	specification	{MKMD.mkmAttribute 40}	–	–		o.4	
42	subordinateObjectClass	{MKMD.mkmAttribute 22}	–	–		o.4	
43	syntaxOrAttribute	{MKMD.mkmAttribute 21}	–	–		o.4	
44	templateDefinition	{MKMD.mkmAttribute 37}	–	–		o.4	
45	templateName	{MKMD.mkmAttribute 7}	–	–		o.4	
46	withAttribute	{MKMD.mkmAttribute 24}	–	–		o.4	
47	withInformationSyntax	{MKMD.mkmAttribute 34}	–	–		o.4	
48	withReplySyntax	{MKMD.mkmAttribute 35}	–	–		o.4	
49	discoveryId	{MKMD.mkmAttribute 41}	–	–		o.4	

Table D.1 (concluded) – Attribute support

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	–		–		–		–		
2	–		–		–		–		
3	–		–		–		–		
4	–		–		–		–		
5	–		–		–		–		
6	–		–		–		–		
7	–		–		–		–		
8	–		–		–		–		
9	–		–		–		–		
10	–		–		–		–		
11	–		–		–		–		
12	–		–		–		–		
13	–		–		–		–		
14	–		–		–		–		
15	–		–		–		–		
16	–		–		–		–		
17	–		–		–		–		
18	–		–		–		–		
19	–		–		–		–		
20	–		–		–		–		
21	–		–		–		–		
22	–		–		–		–		
23	–		–		–		–		
24	–		–		–		–		
25	–		–		–		–		
26	–		–		–		–		
27	–		–		–		–		
28	–		–		–		–		
29	–		–		–		–		
30	–		–		–		–		
31	–		–		–		–		
32	–		–		–		–		
33	–		–		–		–		
34	–		–		–		–		
35	–		–		–		–		
36	–		–		–		–		
37	–		–		–		–		
38	–		–		–		–		
39	–		–		–		–		
40	–		–		–		–		
41	–		–		–		–		
42	–		–		–		–		
43	–		–		–		–		
44	–		–		–		–		
45	–		–		–		–		
46	–		–		–		–		
47	–		–		–		–		
48	–		–		–		–		
49	–		–		–		–		

D.3.2 Actions

The specifier of a manager role implementation that claims to support the actions on the managed objects specified in this Recommendation | International Standard shall import a copy of Table D.2 and complete it.

Table D.2 – Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	mITSearch	{MKMD.mkmAction 1 }	–	o.4		
2	getTextualRepresentation	{MKMD.mkmAction 2 }	–	o.4		

Table D.2 (concluded) – Action support

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	MITSearch (INFORMATION SYNTAX)	–	c:m		
	1.1.1	base	–	c:m		
	1.1.2	discoveryScope	–	c:m		
	1.1.3	classRequest	–	c:o		
	1.2	NameTree (REPLY SYNTAX)	–	c:m		
	1.2.1	rdnInfo	–	c:m		
	1.2.2	subordinates	–	c:m		
2	2.1	AttributeId (INFORMATION SYNTAX)	–	c:m		
	2.2	TextualRepresentation (REPLY SYNTAX)	–	c:m		

Table D.3 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	attributeNotAvailable ViaThisAction	{MKMD.mkmParameter 1 }		c40		
c40: if D.2/2a then m else –						

Annexe E

Formulaire MOCS

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

E.1 Introduction

The purpose of this MOCS proforma is to provide a mechanism for a supplier of an implementation of a Recommendation | International Standard which claims conformance to a managed object class, to provide conformance information in a standard form.

E.2 Instructions for completing the MOCS proforma to produce an MOCS

The MOCS proforma contained in the annex is comprised of information in tabular form, in accordance with ITU-T Rec. X.724 | ISO/IEC 10165-6. The supplier of the implementation shall state which items are supported in the tables that follow and if necessary provide additional information¹⁾.

E.3 Statements of conformance to the managed object classes

E.3.1 Managed object class repertoire managed object class

Table E.1 – Managed object class repertoire managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	managedObjectClass Repertoire	{MKMD.mkmMObjectClass 2}		

If the answer to the actual class question in Table E.1 is No, the supplier of the implementation shall fill in the actual class support Table E.2.

Table E.2 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

¹⁾ Les instructions pour le formulaire MOCS sont spécifiées dans la Rec. UIT-T X.724 | ISO/CEI 10165-6.

Table E.3 – Managed object class repertoire managed object attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63}	–	x		m	
3	packages	{MKMD.dmiAttribute 66}	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50}	–	x		c50	
5	managedObjectClassRepertoireId	{MKMD.mkmAttribute 2}	–	x		m	
6	supportedConditionalPackageList	{MKMD.mkmAttribute 5}	–	x		m	
7	implementedValues	{MKMD.mkmAttribute 6}	–	x		o	

c50: if E.1/1b then – else m

Table E.3 (concluded) – Managed object class repertoire managed object attribute support

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		
6	x		x		x		–		
7	x		x		x		–		

Table E.4 – Notification support

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support		Additional information
					Confirmed	Non-confirmed	
1	attributeValueChange	{MKMD.dmiNotification 1}		m			
2	objectCreation	{MKMD.dmiNotification 6}		m			
3	objectDeletion	{MKMD.dmiNotification 7}		m			

Table E.4 (concluded) – Notification support

Index	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
1	1.1	sourceIndicator	{MKMD.dmiAttribute 26}		o		
	1.2	attributeIdentifierList	{MKMD.dmiAttribute 8}		o		
	1.3	attributeValueChangeDefinition	{MKMD.dmiAttribute 10}		m		
	1.4	notificationIdentifier	{MKMD.dmiAttribute 16}		o		
	1.5	correlatedNotifications	{MKMD.dmiAttribute 12}		o		
	1.6	additionalText	{MKMD.dmiAttribute 7}		o		
	1.7	additionalInformation	{MKMD.dmiAttribute 6}		o		
2	2.1	sourceIndicator	{MKMD.dmiAttribute 26}		o		
	2.2	attributeList	{MKMD.dmiAttribute 9}		o		
	2.3	notificationIdentifier	{MKMD.dmiAttribute 16}		o		
	2.4	correlatedNotifications	{MKMD.dmiAttribute 12}		o		
	2.5	additionalText	{MKMD.dmiAttribute 7}		o		
	2.6	additionalInformation	{MKMD.dmiAttribute 6}		o		
3	3.1	sourceIndicator	{MKMD.dmiAttribute 26}		o		
	3.2	attributeList	{MKMD.dmiAttribute 9}		o		
	3.3	notificationIdentifier	{MKMD.dmiAttribute 16}		o		
	3.4	correlatedNotifications	{MKMD.dmiAttribute 12}		o		
	3.5	additionalText	{MKMD.dmiAttribute 7}		o		
	3.6	additionalInformation	{MKMD.dmiAttribute 6}		o		

E.3.2 Repertoire managed object class

Table E.5 – Repertoire managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	repertoire	{MKMD.mkmMObjectClass 1}		

If the answer to the actual class question in Table E.5 is No, the supplier of the implementation shall fill in the actual class support Table E.6.

Table E.6 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.7 – Package support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	supportedCmipProfilesPackage	{MKMD.mkmPackage 1}	–	o		

Table E.8 – Repertoire managed object attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63}	–	x		m	
3	packages	{MKMD.dmiAttribute 66}	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50}	–	x		c51	
5	repertoireId	{MKMD.mkmAttribute 1}	–	x		m	
6	supportedCmipProfiles	{MKMD.mkmAttribute 4}	–	x		c52	
7	supportedNameBindingList	{MKMD.mkmAttribute 42}	–	x		m	
8	supportedManagedObjectClassList	{MKMD.mkmAttribute 3}	–	x		m	
9	supportedRelationshipClassList	{MKMD.mkmAttribute 44}	–	x		m	
c51: if (not E.5/1b) then m else –							
c52: if (E.7/1a) then m else –							

Table E.8 (concluded) – Repertoire managed object attribute support

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		
6	x		–		–		–		
7	x		x		x		–		
8	x		x		x		–		
9	x		x		x		–		

Table E.9 – Notification support

Index	Notification type template label	Value of object identifier for notification type	Constraints and values	Status	Support		Additional information
					Con- firmed	Non- confirmed	
1	attributeValueChange	{MKMD.dmiNotification 1}		m			
2	objectCreation	{MKMD.dmiNotification 6}		m			
3	objectDeletion	{MKMD.dmiNotification 7}		m			

Table E.9 (concluded) – Notification support

Index	Subindex	Notification field name label	Value of object identifier of attribute type associated with field	Constraints and values	Status	Support	Additional information
1	1.1	sourceIndicator	{MKMD.dmiAttribute 26}		o		
	1.2	attributeIdentifierList	{MKMD.dmiAttribute 8}		o		
	1.3	attributeValueChangeDefinition	{MKMD.dmiAttribute 10}		m		
	1.4	notificationIdentifier	{MKMD.dmiAttribute 16}		o		
	1.5	correlatedNotifications	{MKMD.dmiAttribute 12}		o		
	1.6	additionalText	{MKMD.dmiAttribute 7}		o		
	1.7	additionalInformation	{MKMD.dmiAttribute 6}		o		
2	2.1	sourceIndicator	{MKMD.dmiAttribute 26}		o		
	2.2	attributeList	{MKMD.dmiAttribute 9}		o		
	2.3	notificationIdentifier	{MKMD.dmiAttribute 16}		o		
	2.4	correlatedNotifications	{MKMD.dmiAttribute 12}		o		
	2.5	additionalText	{MKMD.dmiAttribute 7}		o		
	2.6	additionalInformation	{MKMD.dmiAttribute 6}		o		
3	3.1	sourceIndicator	{MKMD.dmiAttribute 26}		o		
	3.2	attributeList	{MKMD.dmiAttribute 9}		o		
	3.3	notificationIdentifier	{MKMD.dmiAttribute 16}		o		
	3.4	correlatedNotifications	{MKMD.dmiAttribute 12}		o		
	3.5	additionalText	{MKMD.dmiAttribute 7}		o		
	3.6	additionalInformation	{MKMD.dmiAttribute 6}		o		

E.3.3 Action template managed object class

Table E.10 – Action template managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	actionTemplate	{MKMD.mkmMObjectClass 12}		

If the answer to the actual class question in Table E.10 is No, the supplier of the implementation shall fill in the actual class support Table E.11.

Table E.11 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.12 – Package support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	templateDefinitionPackage	{MKMD.mkmPackage 3}	–	o		

Table E.13 – Action template managed object attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63}	–	x		m	
3	packages	{MKMD.dmiAttribute 66}	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50}	–	x		c53	
5	templateName	{MKMD.mkmAttribute 7}	–	x		m	
6	templateDefinition	{MKMD.mkmAttribute 37}	–	x		c54	
7	behaviour	{MKMD.mkmAttribute 15}	–	x		m	
8	modeConfirmed	{MKMD.mkmAttribute 33}	–	x		m	
9	parameters	{MKMD.mkmAttribute 13}	–	x		m	
10	withInformationSyntax	{MKMD.mkmAttribute 34}	–	x		m	
11	withReplySyntax	{MKMD.mkmAttribute 35}	–	x		m	
12	registeredAs	{MKMD.mkmAttribute 14}	–	x		m	
c53: if (not E.10/1b) then m else –							
c54: if E.12/1a then m else –							

Table E.13 (concluded) – Action template managed object attribute support

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		
6	x		–		–		–		
7	x		x		x		–		
8	x		–		–		–		
9	x		x		x		–		
10	x		–		–		–		
11	x		–		–		–		
12	x		–		–		–		

Table E.14 – Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	getTextualRepresentation	{MKMD.mkmAction 2}	–	c54		

Table E.14 (concluded) – Action support

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	AttributeId (INFORMATION SYNTAX)	–	c:m		
	1.2	TextualRepresentation (REPLY SYNTAX)	–	c:m		

Table E.15 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	attributeNotAvailable ViaThisAction	{MKMD.mkmParameter 1}		c54		

E.3.4 ASN.1 module managed object class

Table E.16 – ASN.1 module managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	aSN1Module	{MKMD.mkmMObjectClass 4}		

If the answer to the actual class question in Table E.16 is No, the supplier of the implementation shall fill in the actual class support Table E.17.

Table E.17 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.18 – ASN.1 module managed object attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63}	–	x		m	
3	packages	{MKMD.dmiAttribute 66}	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50}	–	x		c55	
5	moduleReference	{MKMD.mkmAttribute 8}	–	x		m	
6	asn1ModuleContents	{MKMD.mkmAttribute 9}	–	x		m	
7	asn1Version	{MKMD.mkmAttribute 45}	–	x		m	
8	registeredAs	{MKMD.mkmAttribute 14}	–	x		m	
c55: if (not E.16/1b) then m else –							

Table E.18 (concluded) – ASN.1 module managed object attribute support

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		
6	x		–		–		–		
7	x		–		–		–		
8	x		–		–		–		

Table E.19 – Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	getTextualRepresentation	{MKMD.mkmAction 2}	–	m		

Table E.19 (concluded) – Action support

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	AttributeId (INFORMATION SYNTAX)	–	m		
	1.2	TextualRepresentation (REPLY SYNTAX)	–	m		

Table E.20 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	attributeNotAvailable ViaThisAction	{MKMD.mkmParameter 1}		m		

E.3.5 Attribute group template managed object class

Table E.21 – Attribute group template managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	attributeGroupTemplate	{MKMD.mkmMObjectClass 10}		

If the answer to the actual class question in Table E.21 is No, the supplier of the implementation shall fill in the actual class support Table E.22.

Table E.22 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.23 – Package support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	templateDefinitionPackage	{MKMD.mkmPackage 3}	–	o		

Table E.24 – Attribute group template managed object attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63}	–	x		m	
3	packages	{MKMD.dmiAttribute 66}	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50}	–	x		c56	
5	templateName	{MKMD.mkmAttribute 7}	–	x		m	
6	templateDefinition	{MKMD.mkmAttribute 37}	–	x		c57	
7	groupElements	{MKMD.mkmAttribute 29}	–	x		m	
8	fixed	{MKMD.mkmAttribute 30}	–	x		m	
9	description	{MKMD.mkmAttribute 31}	–	x		m	
10	registeredAs	{MKMD.mkmAttribute 14}	–	x		m	
c56: if (not E.21/1b) then m else –							
c57: if E.23/1a then m else –							

Table E.24 (concluded) – Attribute group template managed object attribute support

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		
6	x		–		–		–		
7	x		x		x		–		
8	x		–		–		–		
9	x		–		–		–		
10	x		–		–		–		

Table E.25 – Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	getTextualRepresentation	{MKMD.mkmAction 2}	–	c57		

Table E.25 (concluded) – Action support

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	AttributeId (INFORMATION SYNTAX)	–	c:m		
	1.2	TextualRepresentation (REPLY SYNTAX)	–	c:m		

Table E.26 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	attributeNotAvailable ViaThisAction	{MKMD.mkmParameter 1}		c57		

E.3.6 Attribute template managed object class

Table E.27 – Attribute template managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	attributeTemplate	{MKMD.mkmMObjectClass 9}		

If the answer to the actual class question in Table E.27 is No, the supplier of the implementation shall fill in the actual class support Table E.28.

Table E.28 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.29 – Package support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	templateDefinitionPackage	{MKMD.mkmPackage 3}	–	o		

Table E.30 – Attribute template managed object attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63}	–	x		m	
3	packages	{MKMD.dmiAttribute 66}	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50}	–	x		c58	
5	templateName	{MKMD.mkmAttribute 7}	–	x		m	
6	templateDefinition	{MKMD.mkmAttribute 37}	–	x		c59	
7	derivedOrWithSyntaxChoice	{MKMD.mkmAttribute 27}	–	x		m	
8	matchesFor	{MKMD.mkmAttribute 28}	–	x		m	
9	behaviour	{MKMD.mkmAttribute 15}	–	x		m	
10	parameters	{MKMD.mkmAttribute 13}	–	x		m	
11	registeredAs	{MKMD.mkmAttribute 14}	–	x		m	
c58: if (not E.27/1b) then m else –							
c59: if E.29/1a then m else –							

Table E.30 (concluded) – Attribute template managed object attribute support

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		
6	x		–		–		–		
7	x		–		–		–		
8	x		–		–		–		
9	x		x		x		–		
10	x		x		x		–		
11	x		–		–		–		

Table E.31 – Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	getTextualRepresentation	{MKMD.mkmAction 2}	–	c59		

Table E.31 (concluded) – Action support

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	AttributeId (INFORMATION SYNTAX)	–	c:m		
	1.2	TextualRepresentation (REPLY SYNTAX)	–	c:m		

Table E.32 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	attributeNotAvailableViaThisAction	{MKMD.mkmParameter 1}		c59		

E.3.7 Behaviour template managed object class**Table E.33 – Behaviour template managed object class support**

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	behaviourTemplate	{MKMD.mkmMObjectClass 11}		

If the answer to the actual class question in Table E.33 is No, the supplier of the implementation shall fill in the actual class support Table E.34.

Table E.34 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.35 – Package support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	templateDefinitionPackage	{MKMD.mkmPackage 3}	–	o		

Table E.36 – Behaviour template managed object attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63}	–	x		m	
3	packages	{MKMD.dmiAttribute 66}	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50}	–	x		c60	
5	templateName	{MKMD.mkmAttribute 7}	–	x		m	
6	templateDefinition	{MKMD.mkmAttribute 37}	–	x		c61	
7	definedAs	{MKMD.mkmAttribute 32}	–	x		m	
c60: if (not E.33/1b) then m else –							
c61: if E.35/1a then m else –							

Table E.36 (concluded) – Behaviour template managed object attribute support

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		
6	x		–		–		–		
7	x		–		–		–		

Table E.37 – Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	getTextualRepresentation	{MKMD.mkmAction 2}	–	m		

Table E.37 (concluded) – Action support

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	AttributeId (INFORMATION SYNTAX)	–	m		
	1.2	TextualRepresentation (REPLY SYNTAX)	–	m		

Table E.38 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	attributeNotAvailable ViaThisAction	{MKMD.mkmParameter 1}		m		

E.3.8 Document managed object class

Table E.39 – Document managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	document	{MKMD.mkmMObjectClass 14}		

If the answer to the actual class question in Table E.39 is No, the supplier of the implementation shall fill in the actual class support Table E.40.

Table E.40 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.41 – Package support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	documentSpecificationPackage	{MKMD.mkmPackage 4}	–	o		

Table E.42 – Document managed object attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63}	–	x		m	
3	packages	{MKMD.dmiAttribute 66}	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50}	–	x		c62	
5	documentName	{MKMD.mkmAttribute 38}	–	x		m	
6	documentObjectIdentifier	{MKMD.mkmAttribute 39}	–	x		m	
7	specification	{MKMD.mkmAttribute 40}	–	x		c63	

c62: if (not E.39/1b) then m else –
c63: if E.41/1a then m else –

Table E.42 (concluded) – Document managed object attribute support

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		
6	x		–		–		–		
7	x		–		–		–		

Table E.43 – Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	getTextualRepresentation	{MKMD.mkmAction 2}	–	c63		

Table E.43 (concluded) – Action support

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	AttributeId (INFORMATION SYNTAX)	–	c:m		
	1.2	TextualRepresentation (REPLY SYNTAX)	–	c:m		

Table E.44 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	attributeNotAvailableViaThisAction	{MKMD.mkmParameter 1}		c63		

E.3.9 Managed object class template managed object class

Table E.45 – Managed object class template managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	managedObjectClassTemplate	{MKMD.mkmMObjectClass 5}		

If the answer to the actual class question in Table E.45 is No, the supplier of the implementation shall fill in the actual class support Table E.46.

Table E.46 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.47 – Package support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	templateDefinitionPackage	{MKMD.mkmPackage 3}	–	o		

Table E.48 – Managed object class template managed object attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63}	–	x		m	
3	packages	{MKMD.dmiAttribute 66}	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50}	–	x		c64	
5	templateName	{MKMD.mkmAttribute 7}	–	x		m	
6	templateDefinition	{MKMD.mkmAttribute 37}	–	x		c65	
7	derivedFrom	{MKMD.mkmAttribute 10}	–	x		m	
8	characterizedBy	{MKMD.mkmAttribute 11}	–	x		m	
9	conditionalPackages	{MKMD.mkmAttribute 12}	–	x		m	
10	registeredAs	{MKMD.mkmAttribute 14}	–	x		m	
c64: if (not E.45/1b) then m else –							
c65: if E.47/1a then m else –							

Table E.48 (concluded) – Managed object class template managed object attribute support

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		
6	x		–		–		–		
7	x		x		x		–		
8	x		x		x		–		
9	x		x		x		–		
10	x		–		–		–		

Table E.49 – Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	getTextualRepresentation	{MKMD.mkmAction 2}	–	c65		

Table E.49 (concluded) – Action support

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	AttributeId (INFORMATION SYNTAX)	–	c:m		
	1.2	TextualRepresentation (REPLY SYNTAX)	–	c:m		

Table E.50 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	attributeNotAvailable ViaThisAction	{MKMD.mkmParameter 1}		c65		

E.3.10 Name binding template managed object class

Table E.51 – Name binding template managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	nameBindingTemplate	{MKMD.mkmMObjectClass 8}		

If the answer to the actual class question in Table E.51 is No, the supplier of the implementation shall fill in the actual class support Table E.52.

Table E.52 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.53 – Package support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	templateDefinitionPackage	{MKMD.mkmPackage 3}	–	o		

Table E.54 – Name binding template managed object attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63}	–	x		m	
3	packages	{MKMD.dmiAttribute 66}	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50}	–	x		c66	
5	templateName	{MKMD.mkmAttribute 7}	–	x		m	
6	templateDefinition	{MKMD.mkmAttribute 37}	–	x		c67	
7	subordinateObjectClass	{MKMD.mkmAttribute 22}	–	x		m	
8	namedBySuperiorObjectClass	{MKMD.mkmAttribute 23}	–	x		m	
9	withAttribute	{MKMD.mkmAttribute 24}	–	x		m	
10	behaviour	{MKMD.mkmAttribute 15}	–	x		m	
11	create	{MKMD.mkmAttribute 25}	–	x		m	
12	delete	{MKMD.mkmAttribute 26}	–	x		m	
13	registeredAs	{MKMD.mkmAttribute 14}	–	x		m	

c66: if (not E.51/1b) then m else –
c67: if E.53/1a then m else –

Table E.54 (concluded) – Name binding template managed object attribute support

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		
6	x		–		–		–		
7	x		–		–		–		
8	x		–		–		–		
9	x		–		–		–		
10	x		x		x		–		
11	x		–		–		–		
12	x		–		–		–		
13	x		–		–		–		

Table E.55 – Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	getTextualRepresentation	{MKMD.mkmAction 2}	–	c67		

Table E.55 (concluded) – Action support

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	AttributeId (INFORMATION SYNTAX)	–	c:m		
	1.2	TextualRepresentation (REPLY SYNTAX)	–	c:m		

Table E.56 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	attributeNotAvailable ViaThisAction	{MKMD.mkmParameter 1}		c67		

E.3.11 Notification template managed object class

Table E.57 – Notification template managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	notificationTemplate	{MKMD.mkmMObjectClass 13}		

If the answer to the actual class question in Table E.57 is No, the supplier of the implementation shall fill in the actual class support Table E.58.

Table E.58 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.59 – Package support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	templateDefinitionPackage	{MKMD.mkmPackage 3}	–	o		

Table E.60 – Notification template managed object attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63}	–	x		m	
3	packages	{MKMD.dmiAttribute 66}	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50}	–	x		c68	
5	templateName	{MKMD.mkmAttribute 7}	–	x		m	
6	templateDefinition	{MKMD.mkmAttribute 37}	–	x		c69	
7	behaviour	{MKMD.mkmAttribute 15}	–	x		m	
8	parameters	{MKMD.mkmAttribute 13}	–	x		m	
9	withInformationSyntax	{MKMD.mkmAttribute 34}	–	x		m	
10	andAttributeIds	{MKMD.mkmAttribute 36}	–	x		m	
11	withReplySyntax	{MKMD.mkmAttribute 35}	–	x		m	
12	registeredAs	{MKMD.mkmAttribute 14}	–	x		m	
c68: if (not E.57/1b) then m else –							
c69: if E.59/1a then m else –							

Table E.60 (concluded) – Notification template managed object attribute support

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		
6	x		–		–		–		
7	x		x		x		–		
8	x		x		x		–		
9	x		–		–		–		
10	x		x		x		–		
11	x		–		–		–		
12	x		–		–		–		

Table E.61 – Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	getTextualRepresentation	{MKMD.mkmAction 2}	–	c69		

Table E.61 (concluded) – Action support

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	AttributeId (INFORMATION SYNTAX)	–	c:m		
	1.2	TextualRepresentation (REPLY SYNTAX)	–	c:m		

Table E.62 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	attributeNotAvailable ViaThisAction	{MKMD.mkmParameter 1}		c69		

E.3.12 Package template managed object class

Table E.63 – Package template managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	packageTemplate	{MKMD.mkmMObjectClass 6}		

If the answer to the actual class question in Table E.63 is No, the supplier of the implementation shall fill in the actual class support Table E.64.

Table E.64 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.65 – Package support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	templateDefinitionPackage	{MKMD.mkmPackage 3}	–	o		

Table E.66 – Package template managed object attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63}	–	x		m	
3	packages	{MKMD.dmiAttribute 66}	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50}	–	x		c70	
5	templateName	{MKMD.mkmAttribute 7}	–	x		m	
6	templateDefinition	{MKMD.mkmAttribute 37}	–	x		c71	
7	behaviour	{MKMD.mkmAttribute 15}	–	x		m	
8	attributes	{MKMD.mkmAttribute 16}	–	x		m	
9	attributeGroups	{MKMD.mkmAttribute 17}	–	x		m	
10	actions	{MKMD.mkmAttribute 18}	–	x		m	
11	notifications	{MKMD.mkmAttribute 19}	–	x		m	
12	registeredAs	{MKMD.mkmAttribute 14}	–	x		m	
c70: if (not E.63/1b) then m else –							
c71: if E.65/1a then m else –							

Table E.66 (concluded) – Package template managed object attribute support

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		
6	x		–		–		–		
7	x		x		x		–		
8	x		x		x		–		
9	x		x		x		–		
10	x		x		x		–		
11	x		x		x		–		
12	x		–		–		–		

Table E.67 – Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	getTextualRepresentation	{MKMD.mkmAction 2}	–	c71		

Table E.67 (concluded) – Action support

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	AttributeId (INFORMATION SYNTAX)	–	c:m		
	1.2	TextualRepresentation (REPLY SYNTAX)	–	c:m		

Table E.68 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	attributeNotAvailable ViaThisAction	{MKMD.mkmParameter 1}		c71		

E.3.13 Parameter template managed object class

Table E.69 – Parameter template managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	parameterTemplate	{MKMD.mkmMObjectClass 7}		

If the answer to the actual class question in Table E.69 is No, the supplier of the implementation shall fill in the actual class support Table E.70.

Table E.70 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.71 – Package support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	templateDefinitionPackage	{MKMD.mkmPackage 3}	–	o		

Table E.72 – Parameter template managed object attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63}	–	x		m	
3	packages	{MKMD.dmiAttribute 66}	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50}	–	x		c72	
5	templateName	{MKMD.mkmAttribute 7}	–	x		m	
6	templateDefinition	{MKMD.mkmAttribute 37}	–	x		c73	
7	context	{MKMD.mkmAttribute 20}	–	x		m	
8	syntaxOrAttribute	{MKMD.mkmAttribute 21}	–	x		m	
9	behaviour	{MKMD.mkmAttribute 15}	–	x		m	
10	registeredAs	{MKMD.mkmAttribute 14}	–	x		m	
c72: if (not E.69/1b) then m else –							
c73: if E.71/1a then m else –							

Table E.72 (concluded) – Parameter template managed object attribute support)

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		
6	x		–		–		–		
7	x		–		–		–		
8	x		–		–		–		
9	x		x		x		–		
10	x		–		–		–		

Table E.73 – Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	getTextualRepresentation	{MKMD.mkmAction 2}	–	c73		

Table E.73 (concluded) – Action support

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	AttributeId (INFORMATION SYNTAX)	–	c:m		
	1.2	TextualRepresentation (REPLY SYNTAX)	–	c:m		

Table E.74 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	attributeNotAvailableViaThisAction	{MKMD.mkmParameter 1}		c73		

E.3.14 Template managed object class

Table E.75 – Template managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	template	{MKMD.mkmMObjectClass 3}		

If the answer to the actual class question in Table E.75 is No, the supplier of the implementation shall fill in the actual class support Table E.76.

Table E.76 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.77 – Package support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	templateDefinitionPackage	{MKMD.mkmPackage 3}	–	o		

Table E.78 – Template managed object attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63}	–	x		m	
3	packages	{MKMD.dmiAttribute 66}	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50}	–	x		c74	
5	templateName	{MKMD.mkmAttribute 7}	–	x		m	
6	templateDefinition	{MKMD.mkmAttribute 37}	–	x		c75	

c74: if (not E.75/1b) then m else –
c75: if E.77/1a then m else –

Table E.78 (concluded) – Template managed object attribute support

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		
6	x		–		–		–		

Table E.79 – Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	getTextualRepresentation	{MKMD.mkmAction 2}	–	c75		

Table E.79 (concluded) – Action support

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	AttributeId (INFORMATION SYNTAX)	–	c:m		
	1.2	TextualRepresentation (REPLY SYNTAX)	–	c:m		

Table E.80 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	attributeNotAvailableViaThisAction	{MKMD.mkmParameter 1 }		c75		

E.3.15 Discovery managed object class

Table E.81 – Discovery managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	discoveryObject	{MKMD.mkmMObjectClass 15 }		

If the answer to the actual class question in Table E.81 is No, the supplier of the implementation shall fill in the actual class support Table E.82.

Table E.82 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.83 – Discovery managed object attribute support

Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Set by create		Get	
				Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65 }	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63 }	–	x		m	
3	packages	{MKMD.dmiAttribute 66 }	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50 }	–	x		c76	
5	discoveryId	{MKMD.mkmAttribute 41 }	–	x		m	
c76: if (not E.81/1b) then m else –							

Table E.83 (concluded) – Discovery managed object attribute support

Index	Replace		Add		Remove		Set to default		Additional information
	Status	Support	Status	Support	Status	Support	Status	Support	
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		

Table E.84 – Discovery managed object action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	mITSearch	{MKMD.mkmAction 1}	–	m		

Table E.84 (concluded) – Discovery managed object action support

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	MITSearch (INFORMATION SYNTAX)	–			
	1.1.1	base	–	m		
	1.1.2	discoveryScope	–	m		
	1.1.3	classRequest	–	m		
	1.2	NameTree (REPLY SYNTAX)	–			
	1.2.1	rdnInfo	–	m		
	1.2.2	subordinates	–	m		

Annexe F

Formulaire MRCS

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

F.1 Introduction

The purpose of this MRCS proforma for name bindings is to provide a mechanism for a supplier which claims conformance to a name binding to provide conformance information in a standard form.

F.2 Instructions for completing the MRCS proforma for name binding to produce an MRCS

The supplier of the implementation shall state which items are supported in Table F.1 and if necessary provide additional information.²⁾

F.3 Statement of conformance to the name binding

Table F.1 – Name binding support

Index	Name binding template label	Value of object identifier for name binding	Constraints and values	Status	Support	Additional information
1	repertoire-system	{MKMD.mkmNameBinding 1}	One instance shall have repertoireId = "REP1"	o		
2	managedObjectClassRepertoire-repertoire	{MKMD.mkmNameBinding 2}		o		
3	template-document	{MKMD.mkmNameBinding 3}		o		
4	aSN1Module-document	{MKMD.mkmNameBinding 4}		o		
5	discovery-system	{MKMD.mkmNameBinding 5}	discoveryId shall be "DSC1"	o		
6	document-system	{MKMD.mkmNameBinding 6}		o		
7	document-system2	{MKMD.mkmNameBinding 7}		o		

Table F.1 (continued) – Name binding support

Index	Subindex	Operation	Constraints and values	Status	Support	Additional information
1	1.1	Create support		c:o		
	1.1.1	Create with reference object		c:x		
	1.1.2	Create with automatic instance naming		c:o		
	1.2	Delete support		c:o		
	1.2.1	Delete only if no contained objects		c:x		
	1.2.2	Delete contained objects		c:m		
2	2.1	Create support		c:o		
	2.1.1	Create with reference object		c:x		
	2.1.2	Create with automatic instance naming		c:o		

²⁾ Les instructions pour le formulaire MOCS sont spécifiées dans la Rec. UIT-T X.724 | ISO/CEI 10165-6.

Table F.1 (concluded) – Name binding support

Index	Subindex	Operation	Constraints and values	Status	Support	Additional information
	2.2	Delete support		c:o		
	2.2.1	Delete only if no contained objects		c:x		
	2.2.2	Delete contained objects		c:m		
3	3.1	Create support		c:o		
	3.1.1	Create with reference object		c:x		
	3.1.2	Create with automatic instance naming		c:o		
	3.2	Delete support		c:o		
	3.2.1	Delete only if no contained objects		c:x		
	3.2.2	Delete contained objects		c:m		
4	4.1	Create support		c:o		
	4.1.1	Create with reference object		c:x		
	4.1.2	Create with automatic instance naming		c:o		
	4.2	Delete support		c:o		
	4.2.1	Delete only if no contained objects		c:x		
	4.2.2	Delete contained objects		c:m		
5	5.1	Create support		c:o		
	5.1.1	Create with reference object		c:x		
	5.1.2	Create with automatic instance naming		c:o		
	5.2	Delete support		c:o		
	5.2.1	Delete only if no contained objects		c:x		
	5.2.2	Delete contained objects		c:m		
6	6.1	Create support		c:o		
	6.1.1	Create with reference object		c:x		
	6.1.2	Create with automatic instance naming		c:o		
	6.2	Delete support		c:o		
	6.2.1	Delete only if no contained objects		c:x		
	6.2.2	Delete contained objects		c:m		
7	7.1	Create support		c:o		
	7.1.1	Create with reference object		c:x		
	7.1.2	Create with automatic instance naming		c:o		
	7.2	Delete support		c:o		
	7.2.1	Delete only if no contained objects		c:x		
	7.2.2	Delete contained objects		c:m		

Annexe G

Directives pour l'utilisation des objets de connaissance de gestion

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

These guidelines could be used to determine when a managing system gets the management knowledge from management knowledge managed objects and when from management knowledge Directory objects.

Table G.1 – Use of management knowledge objects

	Management knowledge managed objects	Management knowledge Directory objects
Dynamic or static information	Suitable for dynamic information (near real time)	Only for static information
Local or global knowledge	For local knowledge (exists on the managed systems)	Suitable for global knowledge
Access point	A presentation address of the managed system (non-fixed access point)	The well-known presentation address of the Directory (fixed access point)
Knowledge initialization	Management knowledge can be preconfigured in the product by the manufacturer	Management knowledge shall be stored in the Directory system beforehand

Annexe H

Options d'accès pour l'arbre d'annuaire de connaissance des définitions

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

The following options for accessing the Definition Knowledge Directory Tree (DKDT), defined in Annex B, were identified:

- 1) the root of DKDT is identical with the Directory root (this will have an impact on existing first level DSAs);
- 2) via an existing organization (e.g. ANSI) to be used as a "virtual root" for DKDT;
- 3) via a "service provider" type organization that would act as a repository for DKDT (i.e. it would make DKDT available on demand to interested parties).

If option 1 is used, a structure rule like the following is suggested:

```
dkdt1  STRUCTURE-RULE ::= {
      NAME FORM      registeredInformationNameForm
      ID              1}
```

If option 2 or 3 is used, a structure rule like the following is necessary:

```
dkdt2  STRUCTURE-RULE ::= {
      NAME FORM      registeredInformationNameForm
      SUPERIOR RULES{sr2 | sr3 | sr4}
      ID              2}
```

sr2, sr3 and sr4 refer to structure rules defined in Annex B of ITU-T Rec. X.521 | ISO/IEC 9594-7.

To build the definition knowledge Directory tree, the following structure rule is required:

```
dkdt3  STRUCTURE-RULE ::= {
      NAME FORM      registeredInformationNameForm
      SUPERIOR RULES{dkdt1 | dkdt2 | dkdt3}
      ID              3}
```


SÉRIES DES RECOMMANDATIONS UIT-T

- Série A Organisation du travail de l'UIT-T
- Série B Moyens d'expression: définitions, symboles, classification
- Série C Statistiques générales des télécommunications
- Série D Principes généraux de tarification
- Série E Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
- Série F Services de télécommunication non téléphoniques
- Série G Systèmes et supports de transmission, systèmes et réseaux numériques
- Série H Systèmes audiovisuels et multimédias
- Série I Réseau numérique à intégration de services
- Série J Transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
- Série K Protection contre les perturbations
- Série L Construction, installation et protection des câbles et autres éléments des installations extérieures
- Série M Maintenance: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
- Série N Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
- Série O Spécifications des appareils de mesure
- Série P Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
- Série Q Commutation et signalisation
- Série R Transmission télégraphique
- Série S Equipements terminaux de télégraphie
- Série T Terminaux des services télématiques
- Série U Commutation télégraphique
- Série V Communications de données sur le réseau téléphonique
- Série X Réseaux pour données et communication entre systèmes ouverts**
- Série Z Langages de programmation