



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

X.711

(10/97)

SÉRIE X: RÉSEAUX POUR DONNÉES ET
COMMUNICATION ENTRE SYSTÈMES OUVERTS

Gestion OSI – Service et protocole de communication de
gestion

**Technologies de l'information – Interconnexion
des systèmes ouverts – Protocole commun
d'information de gestion: Spécification**

Recommandation UIT-T X.711

(Antérieurement Recommandation du CCITT)

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

RÉSEAUX PUBLICS POUR DONNÉES	
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	
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	
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	
Réseautage	X.600–X.629
Efficacité	X.630–X.639
Qualité de service	X.640–X.649
Dénomination, adressage et enregistrement	X.650–X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680–X.699
GESTION OSI	
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 et fonctions ODMA	X.730–X.799
SÉCURITÉ	X.800–X.849
APPLICATIONS OSI	
Engagement, concomitance et rétablissement	X.850–X.859
Traitement transactionnel	X.860–X.879
Opérations distantes	X.880–X.899
TRAITEMENT RÉPARTI OUVERT	X.900–X.999

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

NORME INTERNATIONALE 9596-1

RECOMMANDATION UIT-T X.711

**TECHNOLOGIES DE L'INFORMATION – INTERCONNEXION DES SYSTÈMES
OUVERTS – PROTOCOLE COMMUN D'INFORMATION
DE GESTION: SPÉCIFICATION**

Source

La Recommandation UIT-T X.711 a été approuvée le 24 octobre 1997. Un texte identique est publié comme Norme internationale ISO/CEI 9596-1.

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. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études à traiter par les Commissions d'études de l'UIT-T, lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la CMNT.

Dans certains secteurs des technologies 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.

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.

DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un Membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

© UIT 1998

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	1
2	1
2.1	1
2.2	2
3	2
3.1	2
3.2	2
3.3	3
3.4	3
3.5	3
3.6	3
4	3
5	4
5.1	4
5.2	4
5.3	5
5.4	5
6	5
6.1	5
6.2	5
6.3	6
6.4	7
6.5	8
6.6	9
6.7	10
6.8	10
6.9	11
6.10	11
7	11
7.1	11
7.2	12
7.3	13
7.4	14
7.5	21
8	22
8.1	22
8.2	23
Annexe A – Règles d'association pour l'élément CMISE	24
A.1 Prescriptions relatives aux services ACSE, session et présentation	24
A.2 Règles d'initialisation de l'association	24
A.3 Règles de terminaison d'association	25
A.4 Règles de rupture d'association	25
Annexe B – Extension de la syntaxe ASN.1	27
Annexe C – Exemples de données APDU d'élément ROSE du service CMISE	35
Annexe D – Syntaxe abstraite d'annuaire	36

NORME INTERNATIONALE

RECOMMANDATION UIT-T

**TECHNOLOGIES DE L'INFORMATION – INTERCONNEXION DES SYSTÈMES
OUVERTS – PROTOCOLE COMMUN D'INFORMATION
DE GESTION: SPÉCIFICATION**

1 Domaine d'application

La présente Recommandation | Norme internationale spécifie un protocole qui est utilisé par des entités de la couche Application pour transférer des informations de gestion.

La présente Recommandation | Norme internationale spécifie:

- les procédures de transmission des informations de gestion entre entités d'application;
- la syntaxe abstraite du protocole commun d'information de gestion (CMIP, *common management information protocol*) et les règles de codage associées à appliquer;
- les procédures d'interprétation des informations de contrôle de protocole;
- les conditions de conformité à remplir par des mises en œuvre de la présente Recommandation | Norme internationale.

La présente Recommandation | Norme internationale ne spécifie pas:

- la structure ou la signification des informations de gestion qui sont transmises au moyen du protocole CMIP;
- la façon dont la gestion est exercée, comme résultat de transferts par protocole CMIP;
- les interactions qui résultent de l'utilisation du protocole CMIP.

2 Références normatives

Les Recommandations et les 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 internationales 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 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.215 (1995) | ISO/CEI 8326:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Définition du service de session*.
- Recommandation UIT-T X.216 (1994) | ISO/CEI 8822:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Définition du service de présentation*.
- 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.226 (1994) | ISO/CEI 8823-1:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Protocole de présentation en mode connexion: spécification du protocole*.

- Recommandation UIT-T X.227 (1995) | ISO/CEI 8650-1:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Protocole en mode connexion applicable à l'élément de service de contrôle d'association: spécification du protocole*.
- Recommandation UIT-T X.710 (1997) | ISO/CEI 9595:1998, *Technologies de l'information – Interconnexion des systèmes ouverts – Service commun de transfert d'informations de gestion*.
- Recommandation UIT-T X.712 (1992) | ISO/CEI 9596-2:1993, *Technologies de l'information – Interconnexion des systèmes ouverts – Protocole commun de transfert d'informations de gestion: formulaire de déclaration de conformité d'une instance de protocole*.

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 des systèmes ouverts – Spécification de la syntaxe abstraite numéro un (ASN.1)*.
- 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 des systèmes ouverts – Spécification des règles de codage de base pour la notation de syntaxe abstraite numéro un (ASN.1)*.
- Recommandation X.219 du CCITT (1988), *Opérations distantes: modèle, notation et définition du service*.
ISO/CEI 9072-1:1989, *Systèmes de traitement de l'information – Communication de texte – Opérations à distance – Partie 1: Modèle, notation et définition du service*.
- Recommandation X.229 du CCITT (1988), *Opérations distantes: spécification du protocole*.
ISO/CEI 9072-2:1989, *Systèmes de traitement de l'information – Communication de texte – Opérations à distance – Partie 2: Spécification du protocole*.
- 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*.

3 Définitions

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

3.1 Définitions relatives au modèle de référence de base

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

- a) élément de service d'application;
- b) processus d'application;
- c) système ouvert réel;
- d) gestion-systèmes.

3.2 Définitions relatives au cadre général de gestion

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. X.700 du CCITT | ISO/CEI 7498-4:

- a) objet géré;
- b) information de gestion;
- c) base d'informations de gestion;
- d) entité d'application de gestion-systèmes.

3.3 Définitions relatives aux opérations distantes

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. X.219 du CCITT | ISO/CEI 9072-1:

- a) initiateur d'association;
- b) répondeur d'association;
- c) opérations liées;
- d) opérations distantes;
- e) élément du service d'opérations distantes;
- f) invocateur;
- g) exécutant;
- h) classe d'associations;
- i) classe d'opérations.

3.4 Définitions relatives au service CMIS

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

- a) attribut;
- b) élément de service commun d'information de gestion;
- c) services communs d'information de gestion;
- d) fournisseur du service CMISE;
- e) utilisateur du service CMISE;
- f) utilisateur invocateur du service CMISE;
- g) utilisateur exécuteur du service CMISE.

3.5 Définitions relatives aux éléments ACSE

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. UIT-T X.217 | ISO/CEI 8649:

- a) contexte d'application;
- b) association d'application;
- c) association.

3.6 Définitions relatives à la présentation

La présente Recommandation | Norme internationale utilise les termes suivants, définis dans la Rec. UIT-T X.216 | ISO/CEI 8822:

- a) syntaxe abstraite;
- b) syntaxe de transfert.

4 Symboles et abréviations

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

ACSE	élément de service de contrôle d'association (<i>association control service element</i>)
APDU	unité de données de protocole d'application (<i>application protocol data unit</i>)
ASE	élément 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'information de gestion (<i>common management information protocol</i>)
CMIPM	machine du protocole commun d'information de gestion (<i>common management information protocol machine</i>)

CMIS	service commun d'information de gestion(<i>common management information service</i>)
CMISE	élément du service commun d'information de gestion (<i>common management information service element</i>)
DCS	ensemble des contextes définis (<i>defined context set</i>)
PCI	information de contrôle du protocole (<i>protocol control information</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>)
RO	opération distante (<i>remote opération</i>)
ROSE	élément de service d'opérations distantes (<i>remote operations service element</i>)
SMAE	entité d'application de gestion-systèmes (<i>systems management application entity</i>)

5 Présentation générale

Le protocole commun d'information de gestion (CMIP) spécifie des éléments de protocole qui peuvent être utilisés pour fournir les services d'opérations et de notifications décrits dans la Rec. UIT-T X.710 | ISO/CEI 9595, qui définit les services communs d'informations de gestion (CMIS, *common management information service*).

5.1 Services fournis

Le protocole spécifié dans la présente Recommandation | Norme internationale assure les services définis dans la Rec. UIT-T X.710 | ISO/CEI 9595. Ces services sont résumés dans le Tableau 1.

Tableau 1 – Services communs d'informations de gestion

Service	Nature	Type
M-CANCEL-GET	Extraction, annulation	Confirmé
M-EVENT-REPORT	Rapport d'événement	Confirmé/non confirmé
M-GET	Extraction	Confirmé
M-SET	Affectation	Confirmé/non confirmé
M-ACTION	Action	Confirmé/non confirmé
M-CREATE	Création	Confirmé
M-DELETE	Suppression	Confirmé

5.2 Services sous-jacents

La présente Recommandation | Norme internationale utilise les services RO-INVOKE (invocation), RO-RESULT (résultat), RO-ERROR (erreur) et RO-REJECT-U (refus par l'utilisateur) de l'élément de service d'opérations distantes (ROSE, *remote operations service element*) qui sont définis dans la Rec. X.219 du CCITT | ISO/CEI 9072-1. L'élément ROSE suppose l'utilisation du service de présentation défini dans la Rec. UIT-T X.216 | ISO/CEI 8822. Les opérations de type «confirmé» du protocole CMIP sont la classe d'opérations 2 (asynchrone) ou la classe d'opérations 1 (synchrone) selon les besoins de l'application. Le choix de la classe d'opérations doit être décidé sur le plan local. Les opérations de type «non confirmé» du protocole CMIP sont celles de la classe d'opérations 5 (asynchrone, résultat non signalé). Le protocole CMIP utilise la classe d'associations 3.

Si l'unité fonctionnelle d'extension de service est adoptée lors de la négociation, les unités de données de protocole d'application ROSE (*ROSE APDU, application protocol data unit*) peuvent être mappées avec d'autres services de présentation que le service P-DATA (transfert de données de présentation).

NOTE – Par exemple, il peut être nécessaire de modifier l'ensemble des contextes définis (DCS, *defined context set*) lorsque l'opération CMIP est envoyée à l'utilisateur de service CMISE équivalent. Dans ce cas, l'unité APDU d'élément ROSE qui achemine l'opération CMIP sera mappée avec le service P-ALTER-CONTEXT (modification de contextes de présentation) qui sert également à apporter les modifications à l'ensemble DCS.

Des détails sur les autres services de présentation requis et sur la façon de les utiliser sont donnés dans la description du contexte d'application en vigueur dans le cadre de l'association.

5.2.1 Services supposés être fournis par l'ACSE

La présente Recommandation | Norme internationale suppose l'utilisation des services suivants: A-ASSOCIATE (association), A-RELEASE (terminaison), A-ABORT (rupture) et A-P-ABORT (rupture par le fournisseur), de l'élément de service de contrôle d'association (ACSE, *association control service element*).

5.2.2 Services supposés être fournis par la couche Présentation

La Rec. X.229 du CCITT | ISO/CEI 9072-2 suppose l'utilisation du service P-DATA de la couche Présentation pour le transfert des unités de données de protocole (PDU, *protocol data unit*) RO-INVOKE, RO-RESULT, RO-ERROR et RO-REJECT APDU.

5.3 Définitions relatives aux informations de gestion

La présente Recommandation | Norme internationale définit la syntaxe abstraite du protocole commun de transfert d'informations de gestion. Les définitions des informations de gestion à acheminer par ce protocole ne sont pas spécifiées dans la présente Recommandation | Norme internationale.

5.4 Version du protocole

La présente Recommandation | Norme internationale définit la version 2 du protocole CMIP. Cette version 2 remplace la version 1. La présente Recommandation | Norme internationale ne définit aucun interfonctionnement entre la version 2 et la version 1.

6 Eléments de procédure

Le présent article donne la définition des éléments de procédure du protocole CMIP. Ces procédures définissent le transfert des unités PDU du protocole CMIP dont la structure, le codage et la relation avec les primitives du service CMIS sont spécifiés à l'article 7.

La machine du protocole commun d'information de gestion (CMIPM, *common management information protocol machine*) reçoit des primitives de service de demande et de réponse du service CMIS et envoie des unités PDU du protocole CMIP en lançant des éléments de procédure spécifiques, comme cela est spécifié dans le présent article.

Une machine CMIPM doit accepter toute unité PDU bien formée du protocole CMIP et la communiquer, pour traitement, à l'utilisateur exécuteur du service CMISE, au moyen des primitives d'indication et de confirmation du service CMIS. Si l'unité PDU reçue n'est pas bien formée ou ne contient pas d'opération ou de notification acceptée, on renvoie une PDU en indiquant que l'unité PDU reçue a été refusée.

Les procédures indiquent uniquement comment interpréter les divers champs de l'unité PDU du protocole CMIP mais non ce qu'un utilisateur invocateur du service CMISE ferait avec les informations qu'il demande, ni comment un utilisateur exécuteur du service CMISE traiterait ces informations.

6.1 Etablissement d'association

L'établissement d'une association implique deux utilisateurs du service CMISE, l'un dans le rôle d'initiateur et l'autre dans celui de répondeur d'association.

Un utilisateur du service CMISE peut lancer un établissement d'association à l'aide du service A-ASSOCIATE de la Rec. UIT-T X.217 | ISO/CEI 8649.

Le contexte d'application précise, notamment, les règles requises pour la coordination de l'information d'initialisation correspondant à différents éléments ASE. Les règles d'association applicables à l'élément CMISE sont spécifiées dans l'Annexe A.

6.2 Opérations distantes

6.2.1 Eléments de procédure RO

Les éléments de procédure du protocole CMIP reposent sur les éléments sous-jacents suivants de la procédure d'opérations distantes:

- a) lancement;
- b) résultat positif;

- c) résultat négatif;
- d) rejet par l'utilisateur;
- e) rejet par le fournisseur.

Ces éléments de procédure sont décrits intégralement dans la Rec. X.229 du CCITT | ISO/CEI 9072-2.

Le Tableau 2 indique la correspondance entre les paramètres CMIS et ROSE.

Tableau 2 – Correspondance entre les paramètres CMIS et ROSE

Paramètre CMIS	Paramètre ROSE
Identificateur d'invocation	ID d'invocation
Identificateur lié	ID lié

La correspondance entre d'autres paramètres CMIS et ROSE est indiquée à l'article 7.

6.2.2 Paramètres du problème de rejet d'opération distante

Les paramètres du problème de rejet d'opération distante (RO-REJECT) sont mappés ou traités comme suit.

6.2.2.1 Le mappage, avec les codes d'erreur du service CMIS, des paramètres du problème d'invocation dû à un rejet par l'utilisateur (RO-REJECT-U) est indiquée dans le Tableau 3.

Tableau 3 – Mappage des paramètres du problème d'invocation RO-REJECT-U avec les codes d'erreur du service CMISE

Paramètres RO-REJECT	Code d'erreur CMISE
Invocation en double	Invocation en double
Argument de type erroné	Argument de type erroné
Limitation de ressources	Limitation de ressources
Opération non reconnue	Opération non reconnue

Les autres paramètres du problème d'invocation dépendent de la réalisation locale.

6.2.2.2 Le traitement des autres paramètres de rejet d'opération distante dépend de la réalisation locale.

6.3 Procédure de notification d'événement

6.3.1 Invocation

Les procédures de notification d'événement sont lancées par la primitive de demande M-EVENT-REPORT.

Dès réception de la primitive de demande M-EVENT-REPORT, la machine CMIPM doit:

- a) dans le mode confirmé, construire une unité APDU demandant l'opération de notification d'événement de gestion de type confirmé (*m-EventReport-Confirmed*) ou, sinon, construire une unité APDU demandant l'opération de notification d'événement de gestion (*m-EventReport*);
- b) envoyer l'unité APDU en appliquant la procédure RO-INVOKE.

6.3.2 Réception

Dès réception d'une unité APDU demandant une opération de notification d'événement de gestion ou de notification d'événement de gestion de type confirmé, la machine CMIPM doit, si l'unité APDU est bien formée, envoyer une primitive d'indication M-EVENT-REPORT à l'utilisateur du service CMISE dont le paramètre de mode indique si la confirmation est demandée ou non; ou, sinon, construire une unité APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure RO-REJECT-U.

6.3.3 Réponse

Dans le mode confirmé, la machine CMIPM doit accepter une primitive de réponse M-EVENT-REPORT et doit:

- a) construire une unité APDU confirmant la notification M-EVENT-REPORT;
- b) si les paramètres présents dans la primitive de réponse M-EVENT-REPORT indiquent que la notification a été acceptée, envoyer l'unité APDU en utilisant la procédure RO-RESULT, sinon envoyer l'unité APDU en utilisant la procédure RO-ERROR.

6.3.4 Réception d'une réponse

Dès réception d'une unité APDU répondant à une notification M-EVENT-REPORT, la machine CMIPM doit, si l'unité APDU est bien formée, envoyer une primitive de confirmation M-EVENT-REPORT à l'utilisateur du service CMISE, terminant ainsi la procédure de notification; ou sinon, construire une unité APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure RO-REJECT-U.

6.4 Procédure d'obtention

6.4.1 Invocation

Les procédures d'extraction sont lancées par la primitive de demande M-GET.

Dès réception de la primitive de demande M-GET, la machine CMIPM doit:

- a) construire une unité APDU demandant l'opération m-Get;
- b) envoyer l'unité APDU en utilisant la procédure RO-INVOKE.

6.4.2 Réception

Dès réception d'une unité APDU demandant l'opération m-Get, la machine CMIPM doit, si l'unité APDU est bien formée, envoyer une primitive d'indication M-GET à l'utilisateur du service CMISE; ou, sinon, construire une unité APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure RO-REJECT-U.

6.4.3 Réponse

La machine CMIPM doit:

- a) accepter zéro, une ou plusieurs primitives de réponse à une demande M-GET contenant un identificateur lié (*linked-ID*), suivies d'une seule primitive de réponse à une demande M-GET sans identificateur lié;
- b) pour chaque primitive de réponse à une demande M-GET contenant un identificateur lié:
 - construire une unité APDU demandant l'opération de réponse liée de gestion (*m-Linked-Reply*) dont l'argument de réponse liée (*LinkedReplyArgument*) est mis comme il se doit sur «erreur de liste d'extraction» (*getListError*), «résultat d'extraction» (*getResult*) ou «défaillance du traitement» (*processingFailure*);
 - envoyer chaque unité APDU en utilisant la procédure RO-INVOKE;
- c) pour la primitive de réponse à une demande M-GET ne contenant pas d'identificateur lié:
 - construire une unité APDU confirmant l'opération de demande m-Get;
 - si les paramètres présents dans la primitive de réponse à une demande M-GET indiquent que l'opération a été exécutée correctement, envoyer l'unité APDU en utilisant la procédure RO-RESULT. Si les paramètres présents dans la primitive de réponse à une demande M-GET indiquent que l'opération n'a obtenu qu'un succès partiel ou n'a pas été exécutée parce qu'il existait une erreur, envoyer l'unité APDU en utilisant la procédure RO-ERROR.

6.4.4 Réception de la réponse

Dès réception d'une unité APDU répondant à une opération m-Get, la machine CMIPM doit:

- a) si l'unité APDU comportait un identificateur lié et qu'elle soit bien formée, envoyer une primitive de confirmation M-GET à l'utilisateur du service CMISE;
- b) si l'unité APDU est la dernière réponse (c'est-à-dire qu'elle ne contient pas d'identificateur lié) et qu'elle soit bien formée, envoyer une primitive de confirmation M-GET à l'utilisateur du service CMISE, terminant ainsi la procédure M-GET;
- c) si l'unité APDU n'est pas bien formée, construire une unité APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure RO-REJECT-U.

6.4.5 Procédure d'extraction, annulation

6.4.5.1 Invocation

Les procédures d'extraction, annulation sont lancées par la primitive de demande M-CANCEL-GET.

Dès réception de la primitive de demande M-CANCEL-GET, la machine CMIPM doit:

- a) construire une unité APDU demandant l'opération d'extraction, annulation de gestion (*m-Cancel-Get*);
- b) envoyer l'unité APDU en utilisant la procédure RO-INVOKE.

6.4.5.2 Réception

Dès réception d'une unité APDU demandant l'opération d'extraction, annulation de gestion, la machine CMIPM doit, si l'unité APDU est bien formée, envoyer une primitive d'indication M-CANCEL-GET à l'utilisateur du service CMISE ou, sinon, construire une unité APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure RO-REJECT-U.

6.4.5.3 Réponse

La machine CMIPM doit:

- a) construire une unité APDU confirmant l'opération d'extraction, annulation de gestion;
- b) si les paramètres contenus dans la primitive de réponse M-CANCEL-GET indiquent que l'opération a été exécutée correctement, envoyer l'unité APDU en faisant appel à la procédure RO-RESULT, sinon envoyer l'unité APDU en faisant appel à la procédure RO-ERROR. Si l'opération *m-Cancel-Get* est bien exécutée, l'utilisateur exécuteur du service CMISE doit cesser d'envoyer des réponses liées vers l'opération *m-Get* et doit envoyer une primitive de réponse M-GET contenant l'erreur «opération annulée».

6.4.5.4 Réception de la réponse

Dès réception d'une unité APDU répondant à une opération d'extraction, annulation de gestion, la machine CMIPM doit, si l'unité APDU est bien formée, envoyer une primitive de confirmation M-CANCEL-GET à l'utilisateur du service CMISE ou, sinon, construire une unité APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure RO-REJECT-U.

6.5 Procédure d'affectation

6.5.1 Invocation

Les procédures d'affectation sont lancées par la primitive de demande M-SET.

Dès réception de la primitive de demande M-SET, la machine CMIPM doit:

- a) dans le mode confirmé, construire une unité APDU demandant l'opération d'affectation de gestion de type confirmé (*m-Set-Confirmed*), ou sinon, construire une unité APDU demandant l'opération d'affectation de gestion (*m-Set*);
- b) envoyer l'unité APDU en utilisant la procédure RO-INVOKE.

6.5.2 Réception

Dès réception d'une unité APDU demandant l'opération d'affectation de gestion ou d'affectation de gestion de type confirmé, la machine CMIPM doit, si l'unité APDU est bien formée, envoyer une primitive d'indication M-SET à l'utilisateur du service CMISE, dont le paramètre de mode indique si une confirmation est demandée ou non; ou, sinon, construire une unité APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure RO-REJECT-U.

6.5.3 Réponse

Dans le mode confirmé, la machine CMIPM doit:

- a) accepter zéro, une ou plusieurs primitives de réponse M-SET contenant un identificateur lié, suivies d'une seule primitive de réponse M-SET sans identificateur lié;
- b) pour chaque primitive de réponse M-SET contenant un identificateur lié:
 - construire une unité APDU demandant l'opération de réponse liée de gestion dont l'argument de réponse liée est mis comme il se doit sur «erreur de liste d'affectation» (*setListError*), «résultat d'affectation» (*setResult*) ou «défaillance du traitement» (*processingFailure*);
 - envoyer chaque unité APDU en utilisant la procédure RO-INVOKE;

- c) pour la primitive de réponse M-SET ne contenant pas d'identificateur lié:
 - construire une unité APDU confirmant l'opération d'affectation de gestion;
 - si les paramètres présents dans la primitive de réponse M-SET indiquent que l'opération a été exécutée correctement, envoyer l'unité APDU en utilisant la procédure RO-RESULT. Si les paramètres présents dans la primitive de réponse M-SET indiquent que l'opération n'a obtenu qu'un succès partiel ou n'a pas été exécutée parce qu'il existait une erreur, envoyer l'unité APDU en utilisant la procédure RO-ERROR.

6.5.4 Réception de la réponse

Dès réception d'une unité APDU répondant à une opération d'affectation de gestion de type confirmé, la machine CMIPM doit:

- a) si l'unité APDU comportait un identificateur lié et qu'elle soit bien formée, envoyer une primitive de confirmation M-SET à l'utilisateur du service CMISE;
- b) si l'unité APDU est la dernière réponse (c'est-à-dire qu'elle ne contient pas d'identificateur lié) et qu'elle soit bien formée, envoyer une primitive de confirmation M-SET à l'utilisateur du service CMISE, terminant ainsi la procédure M-SET;
- c) si l'unité APDU n'est pas bien formée, construire une unité APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure RO-REJECT-U.

6.6 Procédure d'action

6.6.1 Invocation

Les procédures d'action sont lancées par la primitive de demande M-ACTION.

Dès réception de la primitive de demande M-ACTION, la machine CMIPM doit:

- a) dans le mode confirmé, construire une unité APDU demandant l'opération d'action de gestion de type confirmé (*m-Action-Confirmed*); ou sinon, construire une unité APDU demandant l'opération d'action de gestion (*m-Action*);
- b) envoyer l'unité APDU en utilisant la procédure RO-INVOKE.

6.6.2 Réception

Dès réception d'une unité APDU demandant l'opération d'action de gestion ou d'action de gestion de type confirmé, la machine CMIPM doit, si l'unité APDU est bien formée, envoyer une primitive d'indication M-ACTION à l'utilisateur du service CMISE, le paramètre de mode indiquant si une confirmation est demandée ou non ou, sinon, construire une unité APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure RO-REJECT-U.

6.6.3 Réponse

Dans le mode confirmé, la machine CMIPM doit:

- a) accepter zéro, une ou plusieurs primitives de réponse M-ACTION contenant un identificateur lié, suivies d'une primitive de réponse M-ACTION sans identificateur lié;
- b) pour chaque primitive de réponse M-ACTION contenant un identificateur lié:
 - construire une unité APDU demandant l'opération de réponse liée de gestion, l'argument de réponse liée étant mis comme il se doit sur «erreur d'action» (*actionError*), «résultat d'action» (*actionResult*) ou «défaillance du traitement» (*processingFailure*);
 - envoyer chaque unité APDU en utilisant la procédure RO-INVOKE;
- c) pour la primitive de réponse M-ACTION ne contenant pas d'identificateur lié:
 - construire une unité APDU confirmant l'opération d'action de gestion;
 - si les paramètres présents dans la primitive de réponse M-ACTION indiquent que l'opération a été exécutée correctement, envoyer l'unité APDU en utilisant la procédure RO-RESULT, ou sinon, envoyer l'unité APDU en utilisant la procédure RO-ERROR.

6.6.4 Réception de la réponse

Dès réception d'une unité APDU répondant à une opération d'action de gestion de type confirmé, la machine CMIPM doit:

- a) si l'unité APDU comportait un identificateur lié et qu'elle soit bien formée, envoyer une primitive de confirmation M-ACTION à l'utilisateur du service CMISE;
- b) si l'unité APDU est la dernière réponse (c'est-à-dire qu'elle ne contient pas d'identificateur lié) et qu'elle soit bien formée, envoyer une primitive de confirmation M-ACTION à l'utilisateur du service CMISE, terminant ainsi la procédure M-ACTION;
- c) si l'unité APDU n'est pas bien formée, construire une unité APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure RO-REJECT-U.

6.7 Procédure de création

6.7.1 Invocation

Les procédures de création sont lancées par la primitive de demande M-CREATE.

Dès réception de la primitive de demande M-CREATE, la machine CMIPM doit:

- a) construire une unité APDU demandant l'opération de création de gestion (*m-Create*);
- b) envoyer l'unité APDU en utilisant la procédure RO-INVOKE.

6.7.2 Réception

Dès réception d'une unité APDU demandant l'opération de création de gestion, la machine CMIPM doit, si l'unité APDU est bien formée, envoyer une primitive d'indication M-CREATE à l'utilisateur du service CMISE ou sinon, construire une unité APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure RO-REJECT-U.

6.7.3 Réponse

La machine CMIPM doit accepter une primitive de réponse M-CREATE et doit:

- a) construire une unité APDU confirmant l'opération de création de gestion;
- b) si les paramètres présents dans la primitive de réponse M-CREATE indiquent que l'opération a été exécutée correctement, envoyer l'unité APDU en utilisant la procédure RO-RESULT; ou sinon, envoyer l'unité APDU en utilisant la procédure RO-ERROR.

6.7.4 Réception de la réponse

Dès réception d'une unité APDU répondant à une opération de création de gestion, la machine CMIPM doit, si l'unité APDU est bien formée, envoyer une primitive de confirmation M-CREATE à l'utilisateur du service CMISE, terminant ainsi la procédure M-CREATE; ou sinon, construire une unité APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure RO-REJECT-U.

6.8 Procédure de suppression

6.8.1 Invocation

Les procédures de suppression sont lancées par la primitive de demande de M-DELETE.

Dès réception de la primitive de demande M-DELETE, la machine CMIPM doit:

- a) construire une unité APDU demandant l'opération m-Delete;
- b) envoyer l'unité APDU en utilisant la procédure RO-INVOKE.

6.8.2 Réception

Dès réception d'une unité APDU demandant l'opération de suppression de gestion, la machine CMIPM doit, si l'unité APDU est bien formée, envoyer une primitive d'indication M-DELETE à l'utilisateur du service CMISE, ou sinon, construire une unité APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure RO-REJECT-U.

6.8.3 Réponse

La machine CMIPM doit:

- a) accepter zéro, une ou plusieurs primitives de réponse M-DELETE contenant un identificateur lié, suivies d'une seule primitive de réponse M-DELETE sans identificateur lié;
- b) pour chaque primitive de réponse M-DELETE contenant un identificateur lié:
 - construire une unité APDU demandant l'opération de réponse liée de gestion, l'argument de réponse liée étant mis comme il se doit sur «erreur de suppression» (*deleteError*), «résultat de suppression» (*deleteResult*) ou «défaillance du traitement» (*processingFailure*);
 - envoyer chaque unité APDU en utilisant la procédure RO-INVOKE;
- c) pour la primitive de réponse M-DELETE ne contenant pas d'identificateur lié:
 - construire une unité APDU confirmant l'opération de suppression de gestion;
 - si les paramètres présents dans la primitive de réponse M-DELETE indiquent que l'opération a été exécutée correctement, envoyer l'unité APDU en utilisant la procédure RO-RESULT; ou sinon, envoyer l'unité APDU en utilisant la procédure RO-ERROR.

6.8.4 Réception de la réponse

Dès réception d'une unité APDU répondant à une opération de suppression de gestion, la machine CMIPM doit:

- a) si l'unité APDU comportait un identificateur lié et qu'elle soit bien formée, envoyer une primitive de confirmation M-DELETE à l'utilisateur du service CMISE;
- b) si l'unité APDU est la dernière réponse (c'est-à-dire qu'elle ne contient pas d'identificateur lié) et qu'elle soit bien formée, envoyer une primitive de confirmation M-DELETE à l'utilisateur du service CMISE, terminant ainsi la procédure M-DELETE;
- c) si l'unité APDU n'est pas bien formée, construire une unité APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure RO-REJECT-U.

6.9 Terminaison normale d'association

Chacun des utilisateurs du service CMISE peut lancer une terminaison normale d'association en recourant au service A-RELEASE de la Rec. UIT-T X.217 | ISO/CEI 8649.

NOTE – Cette spécification diffère de l'utilisation d'élément ROSE de l'opération BIND (rattachement) dans laquelle seul le demandeur de l'association peut utiliser la procédure A-RELEASE.

6.10 Terminaison brusque d'association

Chacun des utilisateurs du service CMISE peut lancer une terminaison brusque d'association en utilisant le service A-ABORT de la Rec. UIT-T X.217 | ISO/CEI 8649.

L'utilisateur du service CMISE peut recevoir une indication de terminaison brusque de l'association via le service A-ABORT ou A-P-ABORT de la Rec. UIT-T X.217 | ISO/CEI 8649.

7 Syntaxe abstraite¹⁾

Le présent article spécifie la syntaxe abstraite des unités PDU du protocole CMIP.

7.1 Conventions

La syntaxe abstraite est définie à l'aide de la notation décrite dans la Rec. X.208 du CCITT | ISO/CEI 8824. Les productions MACRO de l'ASN.1 utilisées dans la présente Recommandation | Norme internationale ou auxquelles il est fait référence ici ne tiennent pas compte des aspects ambigus de la grammaire.

¹⁾ Les utilisateurs de la présente Recommandation | Norme internationale peuvent librement reproduire les définitions en syntaxe abstraite figurant dans cet article, de façon qu'elles puissent être utilisées aux fins pour lesquelles elles sont prévues.

A chacun des paramètres des services CMISE qui doit être transféré par une PDU du protocole CMIP, correspond un champ d'unité PDU [un type nommé (*Namedtype*) ASN.1], de même nom que ce paramètre (voir la Rec. UIT-T X.710 | ISO/CEI 9595), sous réserve des différences requises par l'utilisation de la notation ASN.1, à savoir que les espaces entre les mots sont supprimés et que la première lettre des mots qui suivent le premier est en majuscule. Par exemple «managed object class» (classe d'objet géré) devient «managedObjectClass». Pour raccourcir certains mots, ceux-ci sont abrégés comme suit:

- ack Accusé de réception (*acknowledgement*)
- arg Argument (*argument*)
- id Identificateur (*identifiant*)
- info Information (*information*)
- sync Synchronisation (*synchronization*)

7.2 Correspondance entre les primitives de l'élément CMISE et les opérations du protocole CMIP

Tableau 4 – Correspondance entre les primitives de l'élément CMISE et les opérations du protocole CMIP

Primitive CMIS	Mode	Identificateur lié	Opération du protocole CMIP
dem./ind. M-CANCEL-GET	Confirmé	Non applicable	Extraction, annulation de gestion de type confirmé (<i>m-Cancel-Get-Confirmed</i>)
rép./conf. M-CANCEL-GET	Non applicable	Non applicable	Extraction, annulation de gestion de type confirmé (<i>m-Cancel-Get-Confirmed</i>)
dem./ind. M-EVENT-REPORT	Non confirmé	Non applicable	Rapport d'événement de gestion (<i>m-EventReport</i>)
dem./ind. M-EVENT-REPORT	Confirmé	Non applicable	Rapport d'événement de gestion de type confirmé (<i>m-EventReport-Confirmed</i>)
rép./conf. M-EVENT-REPORT	Non applicable	Non applicable	Rapport d'événement de gestion de type confirmé (<i>m-EventReport-Confirmed</i>)
dem./ind. M-GET	Confirmé	Non applicable	Extraction de gestion (<i>m-Get</i>)
rép./conf. M-GET	Non applicable	Absent	Extraction de gestion (<i>m-Get</i>)
rép./conf. M-GET	Non applicable	Présent	Réponse liée de gestion (<i>m-Linked-Reply</i>)
dem./ind. M-SET	Non confirmé	Non applicable	Affectation de gestion (<i>m-Set</i>)
dem./ind. M-SET	Confirmé	Non applicable	Affectation de gestion de type confirmé (<i>m-Set-Confirmed</i>)
rép./conf. M-SET	Non applicable	Absent	Affectation de gestion de type confirmé (<i>m-Set-Confirmed</i>)
rép./conf. M-SET	Non applicable	Présent	Réponse liée de gestion (<i>m-Linked-Reply</i>)
dem./ind. M-ACTION	Non confirmé	Non applicable	Action de gestion (<i>m-Action</i>)
dem./ind. M-ACTION	Confirmé	Non applicable	Action de gestion de type confirmé (<i>m-Action-Confirmed</i>)
rép./conf. M-ACTION	Non applicable	Absent	Action de gestion de type confirmé (<i>m-Action-Confirmed</i>)
rép./conf. M-ACTION	Non applicable	Présent	Réponse liée de gestion (<i>m-Linked-Reply</i>)
dem./ind. M-CREATE	Confirmé	Non applicable	Création de gestion (<i>m-Create</i>)
rép./conf. M-CREATE	Non applicable	Non applicable	Création de gestion (<i>m-Create</i>)
dem./ind. M-DELETE	Confirmé	Non applicable	Suppression de gestion (<i>m-Delete</i>)
rép./conf. M-DELETE	Non applicable	Absent	Suppression de gestion (<i>m-Delete</i>)
rép./conf. M-DELETE	Non applicable	Présent	Réponse liée de gestion (<i>m-Linked-Reply</i>)

NOTE – Le mappage des macros OPERATION et ERROR avec l'élément ROSE est défini dans la Rec. X.219 du CCITT | ISO/CEI 9072-1.

7.3 Données d'utilisateur d'élément ACSE

Le protocole ACSE (Rec. UIT-T X.227 | ISO/CEI 8650-1) est décrit en notation ASN.1. Les «informations d'utilisateur» sont décrites en utilisant le type de données EXTERNAL.

7.3.1 Données d'utilisateur de la primitive A-ASSOCIATE

Le codage des informations d'utilisateur du protocole CMIP, à insérer dans la primitive A-ASSOCIATE par le paramètre «informations d'utilisateur» est défini comme suit:

CMIP-A-ASSOCIATE-Information {joint-iso-itu-t ms(9) cmip(1) modules(0) aAssociateUserInfo(1)}

DEFINITIONS ::= BEGIN

FunctionalUnits ::= BIT STRING {
multipleObjectSelection (0),
filter (1),
multipleReply (2),
extendedService (3),
cancelGet (4)
}

-- L'unité fonctionnelle *i* n'est prise en charge que si le bit *i* est égal à un

-- L'information est acheminée dans le paramètre "informations d'utilisateur" de la primitive A-ASSOCIATE

CMIPUserInfo ::= SEQUENCE {

protocolVersion [0] IMPLICIT ProtocolVersion DEFAULT { version1 },
functionalUnits [1] IMPLICIT FunctionalUnits DEFAULT {},
accessControl [2] EXTERNAL OPTIONAL,
userInfo [3] EXTERNAL OPTIONAL
}

ProtocolVersion ::= BIT STRING {

version1 (0),
version2 (1)
}

END

Le codage des autres "informations d'utilisateur" fournies par l'utilisateur du service CMISE n'est pas défini dans la présente Recommandation | Norme internationale.

7.3.2 Données d'utilisateur de la primitive A-ABORT

Le codage des informations d'utilisateur CMIP à communiquer à la primitive A-ABORT dans le paramètre "informations d'utilisateur" est défini comme suit:

CMIP-A-ABORT-Information {joint-iso-itu-t ms(9) cmip(1) modules(0) aAbortUserInfo(2)}

DEFINITIONS ::= BEGIN

-- Information acheminée dans le paramètre "informations d'utilisateur" de la primitive A-ABORT

CMIPAbortInfo ::= SEQUENCE {

abortSource [0] IMPLICIT CMIPAbortSource,
userInfo [1] EXTERNAL OPTIONAL
}

CMIPAbortSource ::= ENUMERATED {

cmiseServiceUser (0),
cmiseServiceProvider (1)
}

END

Le codage des autres "informations d'utilisateur" fourni par l'utilisateur du service CMISE n'est pas défini dans la présente Recommandation | Norme internationale.

7.4 Unités de données CMIP

Le protocole est décrit en terme d'unités de données du protocole commun d'information de gestion, échangées entre CMISE homologues. Les unités PDU sont spécifiées en utilisant la notation ASN.1 et les macro externes OPERATION et ERROR du protocole d'opérations distantes, définies dans la Rec. X.219 du CCITT | ISO/CEI 9072-1.

-- *Protocole commun d'information de gestion (CMIP)*

CMIP-1 {joint-iso-itu-t ms(9) cmip(1) modules(0) protocol(3)}

DEFINITIONS ::= **BEGIN**

-- *Définition des opérations distantes*

IMPORTS OPERATION, ERROR FROM Remote-Operation-Notation {joint-iso-ccitt remote-operations(4) notation(0)}

-- *Définition du service d'opérations distantes*

InvokeIDType FROM Remote-Operations-APDUs {joint-iso-ccitt remote-operations(4) apdus(1)}

-- *Définition du service d'annuaire*

-- *La présente Recommandation / Norme internationale reprend la syntaxe abstraite de la Rec. X.501 du CCITT (1988) / ISO/CEI 9594-2:1990. Un extrait de celle-ci, suffisant pour répondre aux besoins du protocole CMIP, figure dans l'Annexe D de la présente Recommandation / Norme internationale .*

DistinguishedName, RDNSSequence

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

-- *Opérations CMISE*

-- *Dans les opérations ci-après, le type d'argument est obligatoire dans l'unité APDU d'élément ROSE correspondant*

-- *Opérations d'action (M-ACTION)*

m-Action OPERATION

ARGUMENT ActionArgument

::= **localValue** : 6

m-Action-Confirmed OPERATION

ARGUMENT ActionArgument

RESULT ActionResult -- *Ce résultat est conditionnel; pour les conditions, voir 8.3.3.2.9 de la Rec. UIT-T X.710 | ISO/CEI 9595*

ERRORS { **accessDenied, classInstanceConflict, complexityLimitation, invalidScope, invalidArgumentValue, invalidFilter, noSuchAction, noSuchArgument, noSuchObjectClass, noSuchObjectInstance, processingFailure, syncNotSupported** }

LINKED { **m-Linked-Reply** }

::= **localValue** : 7

m-CancelGet OPERATION

ARGUMENT

getInvokeId InvokeIDType

RESULT

ERRORS { **mistypedOperation, noSuchInvokeId, processingFailure** }

::= **localValue** : 10

-- *Opération de création (M-CREATE)*

m-Create OPERATION

ARGUMENT CreateArgument

RESULT CreateResult -- *Ce résultat est conditionnel; pour les conditions, voir 8.3.4.1.3 de la Rec. UIT-T X.710 | ISO/CEI 9595*

ERRORS { **accessDenied, classInstanceConflict, duplicateManagedObjectInstance, invalidAttributeValue, invalidObjectInstance, missingAttributeValue, noSuchAttribute, noSuchObjectClass, noSuchObjectInstance, noSuchReferenceObject, processingFailure** }

::= **localValue** : 8

-- *Opération de suppression (M-DELETE)*

m-Delete OPERATION

Argument DeleteArgument

RESULT DeleteResult -- *Ce résultat est conditionnel; pour les conditions, voir 8.3.5.2.8 de la Rec. UIT-T X.710 | ISO/CEI 9595*

ERRORS { **accessDenied, classInstanceConflict, complexityLimitation, invalidFilter, InvalidScope, noSuchObjectClass, noSuchObjectInstance, processingFailure, syncNotSupported** }

LINKED { m-Linked-Reply }
::= localValue : 9

-- *Opération de rapport d'événement (M-EVENT-REPORT)*

m-EventReport OPERATION

ARGUMENT **EventReportArgument**
::= localValue : 0

m-EventReport-Confirmed OPERATION

ARGUMENT **EventReportArgument**
RESULT **EventReportResult** -- *en option*
ERRORS { **invalidArgumentValue, noSuchArgument, noSuchEventType, noSuchObjectClass,**
 noSuchObjectInstance, processingFailure }
::= localValue : 1

-- *Opération d'extraction (M-GET)*

m-Get OPERATION

ARGUMENT **GetArgument**
RESULT **GetResult** -- *Ce résultat est conditionnel; pour les conditions, voir 8.3.1.2.8 de la*
 -- *Rec. UIT-T X.710 | ISO/CEI 9595*
ERRORS { **accessDenied, classInstanceConflict, complexityLimitation, getListError, invalidFilter, invalidScope,**
 noSuchObjectClass, noSuchObjectInstance, operationCancelled, processingFailure, syncNotSupported
 }
LINKED { m-Linked-Reply }
::= localValue : 3

-- *Opération liée pour M-GET, M-SET (de type confirmé), M-ACTION (de type confirmé) et M-DELETE*

m-Linked-Reply OPERATION

ARGUMENT **LinkedReplyArgument**
::= localValue : 2

-- *Opération d'affectation (M-SET)*

m-Set OPERATION

ARGUMENT **SetArgument**
::= localValue : 4

m-Set-Confirmed OPERATION

ARGUMENT **SetArgument**
RESULT **SetResult** -- *Ce résultat est conditionnel; pour les conditions, voir 8.3.2.2.9 de la*
 -- *Rec. UIT-T X.710 | ISO/CEI 9595*
ERRORS { **accessDenied, classInstanceConflict, complexityLimitation, invalidFilter, invalidScope,**
 noSuchObjectClass, noSuchObjectInstance, processingFailure, setListError, syncNotSupported }
LINKED { m-Linked-Reply }
::= localValue : 5

-- *Définitions d'erreur du service CMIS*

-- *Dans les erreurs ci-après, sauf indication contraire, le type de paramètre est obligatoire dans l'unité APDU*

-- *d'élément ROSE correspondant*

accessDenied ERROR

::= localValue : 2

classInstanceConflict ERROR

PARAMETER **BaseManagedObjectId**
::= localValue : 19

complexityLimitation ERROR

PARAMETER **ComplexityLimitation**-- *en option*
::= localValue : 20

duplicateManagedObjectInstance ERROR

PARAMETER **ObjectInstance**
::= localValue : 11

getListError ERROR

PARAMETER **GetListError**
::= localValue : 7

invalidArgumentValue ERROR

PARAMETER InvalidArgumentValue
 ::= localValue : 15

invalidAttributeValue ERROR

PARAMETER Attribute
 ::= localValue : 6

invalidFilter ERROR

PARAMETER CMISFilter
 ::= localValue : 4

invalidObjectInstance ERROR

PARAMETER ObjectInstance
 ::= localValue : 17

invalidScope ERROR

PARAMETER Scope
 ::= localValue : 16

missingAttributeValue ERROR

PARAMETER SET OF AttributeId
 ::= localValue : 18

mistypedOperation ERROR

::= localValue : 21

noSuchAction ERROR

PARAMETER NoSuchAction
 ::= localValue : 9

noSuchArgument ERROR

PARAMETER NoSuchArgument
 ::= localValue : 14

noSuchAttribute ERROR

PARAMETER AttributeId
 ::= localValue : 5

noSuchEventType ERROR

PARAMETER NoSuchEventType
 ::= localValue : 13

noSuchInvokeId ERROR

PARAMETER InvokeIDType
 ::= localValue : 22

noSuchObjectClass ERROR

PARAMETER ObjectClass
 ::= localValue : 0

noSuchObjectInstance ERROR

PARAMETER ObjectInstance
 ::= localValue : 1

noSuchReferenceObject ERROR

PARAMETER ObjectInstance
 ::= localValue : 12

operationCancelled ERROR

::= localValue : 23

processingFailure ERROR

PARAMETER ProcessingFailure -- en option
 ::= localValue : 10

setListError ERROR

PARAMETER SetListError
 ::= localValue : 8

syncNotSupported ERROR

PARAMETER CMISync
 ::= localValue : 3

-- Définitions des types supports

AttributeId ::= CHOICE {

globalForm [0] IMPLICIT OBJECT IDENTIFIER,
localForm [1] IMPLICIT INTEGER
}

- La présente Recommandation / Norme internationale n'affecte aucune valeur à localForm. Lorsque cette solution
- est adoptée, les valeurs admissibles pour les nombres entiers ainsi que leurs significations doivent être définies dans
- le cadre du contexte d'application propre à leur utilisation

AttributeIdError ::= SEQUENCE {

errorStatus ENUMERATED {
accessDenied (2),
noSuchAttribute (5),
attributeId AttributeId
}

BaseManagedObjectId ::= SEQUENCE {

baseManagedObjectClass ObjectClass,
baseManagedObjectInstance ObjectInstance
}

CMISFilter ::= CHOICE {

item [8] FilterItem,
and [9] IMPLICIT SET OF CMISFilter,
or [10] IMPLICIT SET OF CMISFilter,
not [11] CMISFilter
}

CMISSync ::= ENUMERATED {

bestEffort (0),
atomic (1) }

ComplexityLimitation ::= SET {

scope [0] Scope OPTIONAL,
filter [1] CMISFilter OPTIONAL,
sync [2] CMISSync OPTIONAL
}

CreateArgument ::= SEQUENCE {

managedObjectClass ObjectClass,
managedOrSuperiorObjectInstance CHOICE {
managedObjectInstance ObjectInstance,
superiorObjectInstance [8] ObjectInstance } OPTIONAL,
accessControl [5] AccessControl OPTIONAL,
referenceObjectInstance [6] ObjectInstance OPTIONAL,
attributeList [7] IMPLICIT SET OF Attribute OPTIONAL
}

CreateResult ::= SEQUENCE {

managedObjectClass ObjectClass OPTIONAL,
managedObjectInstance ObjectInstance OPTIONAL, -- doit être retourné si omis de CreateArgument
currentTime [5] IMPLICIT GeneralizedTime OPTIONAL,
attributeList [6] IMPLICIT SET OF Attribute OPTIONAL
}

DeleteArgument ::= SEQUENCE {

COMPONENTS OF BaseManagedObjectId,
accessControl [5] AccessControl OPTIONAL,
synchronization [6] IMPLICIT CMISSync DEFAULT bestEffort,
scope [7] Scope DEFAULT namedNumbers : baseObject,
filter CMISFilter DEFAULT and : { }
}

DeleteError ::= SEQUENCE {

managedObjectClass ObjectClass OPTIONAL,
managedObjectInstance ObjectInstance OPTIONAL,
currentTime [5] IMPLICIT GeneralizedTime OPTIONAL,
deleteErrorInfo [6] ENUMERATED { accessDenied (2) }
}

DeleteResult ::= SEQUENCE {

managedObjectClass ObjectClass OPTIONAL,
managedObjectInstance ObjectInstance OPTIONAL,
currentTime [5] IMPLICIT GeneralizedTime OPTIONAL
}

```

EventReply ::= SEQUENCE {
    eventType      EventTypeId,
    eventReplyInfo [8] ANY DEFINED BY eventType OPTIONAL
}

```

```

EventReportArgument ::= SEQUENCE {
    managedObjectClass      ObjectClass,
    managedObjectInstance   ObjectInstance,
    eventTime                [5] IMPLICIT GeneralizedTime OPTIONAL,
    eventType                EventTypeId,
    eventInfo                [8] ANY DEFINED BY eventType OPTIONAL
}

```

```

EventReportResult ::= SEQUENCE {
    managedObjectClass      ObjectClass OPTIONAL,
    managedObjectInstance   ObjectInstance OPTIONAL,
    currentTime            [5] IMPLICIT GeneralizedTime OPTIONAL,
    eventReply              EventReply OPTIONAL
}

```

```

EventTypeId ::= CHOICE {
    globalForm      [6] IMPLICIT OBJECT IDENTIFIER,
    localForm       [7] IMPLICIT INTEGER
}

```

-- La présente Recommandation / Norme internationale n'affecte aucune valeur à localForm. Lorsque cette solution est adoptée, les valeurs admissibles pour les nombres entiers ainsi que leurs significations doivent être définies dans le cadre du contexte d'application propre à leur utilisation

```

FilterItem ::= CHOICE {
    equality          [0] IMPLICIT Attribute,
    substrings       [1] IMPLICIT SEQUENCE OF CHOICE {
        initialString [0] IMPLICIT SEQUENCE {
            attributeIdAttributeId,
            string      ANY DEFINED BY attributeId },
        anyString      [1] IMPLICIT SEQUENCE {
            attributeIdAttributeId,
            string      ANY DEFINED BY attributeId },
        finalString    [2] IMPLICIT SEQUENCE {
            attributeIdAttributeId,
            string      ANY DEFINED BY attributeId },
    greaterOrEqual   [2] IMPLICIT Attribute, -- valeur retenue ≥ valeur de l'attribut
    lessOrEqual      [3] IMPLICIT Attribute, -- valeur retenue ≤ valeur de l'attribut
    present          [4] AttributeId,
    subsetOf         [5] IMPLICIT Attribute, -- valeur retenue: sous-ensemble de la valeur de l'attribut
    supersetOf       [6] IMPLICIT Attribute, -- valeur retenue: sous-ensemble de la valeur de l'attribut
    nonNullSetIntersection [7] IMPLICIT Attribute
}

```

```

GetArgument ::= SEQUENCE {
    COMPONENTS OF      BaseManagedObjectId,
    accessControl       [5] AccessControl OPTIONAL,
    synchronization    [6] IMPLICIT CMISync DEFAULT bestEffort,
    scope              [7] Scope DEFAULT namedNumbers : baseObject,
    filter              CMISFilter DEFAULT and : {},
    attributeIdList     [12] IMPLICIT SET OF AttributeId OPTIONAL
}

```

```

GetInfoStatus ::= CHOICE {
    attributeIdError    [0] IMPLICIT AttributeIdError,
    attribute           [1] IMPLICIT Attribute
}

```

```

GetListError ::= SEQUENCE {
    managedObjectClass      ObjectClass OPTIONAL,
    managedObjectInstance   ObjectInstance OPTIONAL,
    currentTime            [5] IMPLICIT GeneralizedTime OPTIONAL,
    getInfoList             [6] IMPLICIT SET OF GetInfoStatus
}

```

```

GetResult ::= SEQUENCE {
    managedObjectClass ObjectClass OPTIONAL,
    managedObjectInstance ObjectInstance OPTIONAL,
    currentTime [5] IMPLICIT GeneralizedTime OPTIONAL,
    attributeList [6] IMPLICIT SET OF Attribute OPTIONAL
}

```

```

InvalidArgumentValue ::= CHOICE {
    actionValue [0] IMPLICIT ActionInfo,
    eventValue [1] IMPLICIT SEQUENCE {
        eventType EventTypeId,
        eventInfo [8] ANY DEFINED BY eventType OPTIONAL
    }
}

```

```

LinkedReplyArgument ::= CHOICE {
    getResult [0] IMPLICIT GetResult,
    getListError [1] IMPLICIT GetListError,
    setResult [2] IMPLICIT SetResult,
    setListError [3] IMPLICIT SetListError,
    actionResult [4] IMPLICIT ActionResult,
    processingFailure [5] IMPLICIT ProcessingFailure,
    deleteResult [6] IMPLICIT DeleteResult,
    actionError [7] IMPLICIT ActionError,
    deleteError [8] IMPLICIT DeleteError
}

```

```

ModifyOperator ::= INTEGER {
    replace (0),
    addValues (1),
    removeValues (2),
    setToDefault (3)
}

```

```

NoSuchAction ::= SEQUENCE {
    managedObjectClass ObjectClass,
    actionType ActionTypeId
}

```

```

NoSuchArgument ::= CHOICE {
    actionId [0] IMPLICIT SEQUENCE {
        managedObjectClass ObjectClass OPTIONAL,
        actionType ActionTypeId
    },
    eventId [1] IMPLICIT SEQUENCE {
        managedObjectClass ObjectClass OPTIONAL,
        eventType EventTypeId
    }
}

```

```

NoSuchEventType ::= SEQUENCE {
    managedObjectClass ObjectClass,
    eventType EventTypeId
}

```

```

ObjectClass ::= CHOICE {
    globalForm [0] IMPLICIT OBJECT IDENTIFIER,
    localForm [1] IMPLICIT INTEGER
}

```

- La présente Recommandation / Norme internationale n'affecte aucune valeur à localForm. Lorsque cette solution
- est adoptée, les valeurs admissibles pour les nombres entiers ainsi que leurs significations doivent être définies dans
- le cadre du contexte d'application propre à leur utilisation

```

ObjectInstance ::= CHOICE {
    distinguishedName [2] IMPLICIT DistinguishedName,
    nonSpecificForm [3] IMPLICIT OCTET STRING,
    localDistinguishedName [4] IMPLICIT RDNSSequence
}

```

- localDistinguishedName est la partie du nom distinctif qui est nécessaire pour identifier sans ambiguïté l'objet géré
- dans le cadre de la communication entre les systèmes ouverts

```

ProcessingFailure ::= SEQUENCE {
    managedObjectClass ObjectClass,
    managedObjectInstance ObjectInstance OPTIONAL,
    specificErrorInfo [5] SpecificErrorInfo
}

```

```

Scope ::= CHOICE { namedNumbers INTEGER { baseObject (0),
                                             firstLevelOnly (1),
                                             wholeSubtree (2) },
  individualLevels [1] IMPLICIT INTEGER, -- Le nombre entier POSITIF indique le niveau à choisir
  baseToNthLevel [2] IMPLICIT INTEGER } -- Le nombre entier POSITIF N indique qu'il faut opérer un choix
                                         -- dans la gamme de niveaux (0 – N)
-- Avec individualLevels et baseToNthLevel, une valeur 0 a la même sémantique que baseObject
-- avec individualLevels, une valeur 1 a la même sémantique que firstLevelOnly

SetArgument ::= SEQUENCE {
  COMPONENTS OF BaseManagedObjectId,
  accessControl [5] AccessControl OPTIONAL,
  synchronization [6] IMPLICIT CMISync DEFAULT bestEffort,
  scope [7] Scope DEFAULT namedNumbers : baseObject,
  filter CMISFilter DEFAULT and : {},
  modificationList [12] IMPLICIT SET OF SEQUENCE {
    modifyOperator [2] IMPLICIT ModifyOperator DEFAULT replace,
    attributeId AttributeId,
    attributeValue ANY DEFINED BY attributeId OPTIONAL -- peut être absent dans setToDefault
  }
}

SetInfoStatus ::= CHOICE {
  attributeError [0] IMPLICIT AttributeError,
  attribute [1] IMPLICIT Attribute
}

SetListError ::= SEQUENCE {
  managedObjectClass ObjectClass OPTIONAL,
  managedObjectInstance ObjectInstance OPTIONAL,
  currentTime [5] IMPLICIT GeneralizedTime OPTIONAL,
  setInfoList [6] IMPLICIT SET OF SetInfoStatus
}

SetResult ::= SEQUENCE {
  managedObjectClass ObjectClass OPTIONAL,
  managedObjectInstance ObjectInstance OPTIONAL,
  currentTime [5] IMPLICIT GeneralizedTime OPTIONAL,
  attributeList [6] IMPLICIT SET OF Attribute OPTIONAL
}

SpecificErrorInfo ::= SEQUENCE {
  errorId OBJECT IDENTIFIER,
  errorInfo ANY DEFINED BY errorId
}

END -- Fin des définitions de la syntaxe du protocole CMIP

```

7.5 Définition de syntaxe abstraite pour le protocole CMIP

La présente Recommandation | Norme internationale affecte la valeur de l'identificateur d'objet ASN.1

```
{joint-iso-itu-t ms(9) cmip(1) cmip-pci(1) abstractSyntax(4)}
```

comme nom de syntaxe abstraite pour l'ensemble des valeurs de données de présentation, dont chacune est soit une valeur de type ASN.1:

Remote-Operations-APDUs.ROSEapdus

comme définie dans la Rec. X.229 du CCITT | ISO/CEI 9072-2, le composant argument étant rempli selon les définitions de la version de protocole CMIP-1, ou une valeur de l'un des types ASN.1 suivants:

- **CMIP-A-ASSOCIATE-Information.CMIPUserInfo;**
- **CMIP-A-ABORT-Information.CMIPAbortInfo.**

La valeur de descripteur d'objet ASN.1 correspondante peut être:

"CMIP-PCI".

Cette syntaxe abstraite est définie en vue d'inclure tous les types de données résolus par les productions ANY DEFINED BY X, dans lesquelles X est du type OBJECT IDENTIFIER.

Les valeurs d'identificateur d'objet et de descripteur d'objet ASN.1:

{joint-iso-ccitt asn1(1) basic-encoding(1)} et "Basic Encoding of single ASN.1 type"

(affectées à un objet dans la Rec. X.209 du CCITT | ISO/CEI 8825) peuvent être utilisées comme nom de syntaxe de transfert avec cette syntaxe abstraite.

7.5.1 Règles d'extensibilité

7.5.1.1 La machine CMIPM acceptrice doit, lorsqu'elle traite une information de type CMIP-A-ASSOCIATE:

- ne tenir compte d'aucune valeur d'étiqueté non définie dans la syntaxe abstraite de la présente Recommandation | Norme internationale; et
- ne tenir compte d'aucune affectation de nom binaire inconnu dans une chaîne binaire (BIT STRING).

7.5.1.2 Le nom de syntaxe abstraite peut être utilisé lorsque les valeurs des données de présentation sont modifiées de façon à inclure:

- de nouvelles opérations de gestion-systèmes;
- de nouveaux éléments étiquetés dans un ensemble ou dans une séquence (SET ou SEQUENCE);
- de nouvelles affectations de nom binaire dans une chaîne binaire (BIT STRING); et
- de nouveaux nombres nommés pour un entier (INTEGER); et
- de nouvelles énumérations nommées dans un énuméré (ENUMERATED).

8 Conformité

Un système revendiquant la mise en œuvre des procédures spécifiées dans la présente Recommandation | Norme internationale doit se conformer aux spécifications des 8.1 et 8.2.

8.1 Conditions de conformité statique

Une mise en œuvre revendiquant la conformité à la présente Recommandation | Norme internationale doit être assortie d'une déclaration PICS conforme au formulaire de déclaration PICS spécifié dans la Rec. UIT-T X.712 | ISO/CEI 9596-2. Cette déclaration doit en particulier préciser que la mise en œuvre:

- 1) soit, prend en charge le protocole nécessaire pour fournir tous les services CMIS définis dans l'unité fonctionnelle "noyau", dans les deux rôles d'invocateur et d'exécuteur;
- 2) soit, prend en charge le protocole nécessaire pour fournir certains des services CMIS définis dans l'unité fonctionnelle "noyau", dans les rôles d'invocateur et/ou d'exécuteur, mais non tous les protocoles dans les deux rôles pour tous les services CMIS.

La mise en œuvre doit:

- a) prendre en charge les éléments de procédure spécifiés dans l'article 6 de la présente Recommandation | Norme internationale pour les services utilisés dans le rôle pour lequel la conformité est revendiquée;
- b) prendre en charge la syntaxe abstraite associée aux unités de données de protocole qui sont nécessaires pour acheminer les requêtes et/ou les demandes concernant ces services dans le rôle pour lequel la conformité est revendiquée;
- c) prendre en charge la syntaxe de transfert dérivée des règles de codage spécifiées dans la Rec. X.209 du CCITT | ISO/CEI 8825 et nommée:

{joint-iso-ccitt asn1(1) basic-encoding(1)},

afin de générer et d'interpréter les unités PDU de protocole CMIP, telles qu'elles sont définies par la syntaxe abstraite:

"CMIP-PCI",

pour les éléments de protocole avec lesquels la conformité est revendiquée:

- d) prendre en charge le protocole ACSE défini dans la Rec. UIT-T X.227 | ISO/CEI 8650-1 pour établir et terminer une association;
- e) prendre en charge les règles spécifiées dans l'Annexe A dans tout contexte d'application comptant l'élément CMISE parmi l'un des éléments ASE;

- f) prendre en charge la classe d'associations 3 du protocole d'élément ROSE défini dans la Rec. X.229 du CCITT | ISO/CEI 9072-2;
- g) prendre en charge les éléments de protocole requis pour fournir l'unité fonctionnelle de réponses multiples si l'unité fonctionnelle de sélection d'objets multiples est sélectionnée;
- h) prendre en charge les éléments de protocole requis pour fournir le service M-GET si l'unité fonctionnelle d'obtention d'annulation est prise en charge.

8.2 Conditions de conformité dynamique

Le système doit:

- a) prendre en charge les éléments de procédure pour chaque service CMIS dans le rôle pour lequel la conformité est revendiquée;
- b) s'ils sont utilisés, vérifier les paramètres de sécurité facultatifs définis dans les unités PDU du protocole CMIP;
- c) si l'unité fonctionnelle d'extension de services est prise en charge, prendre en charge le protocole de présentation défini dans la Rec. UIT-T X.226 | ISO/CEI 8823-1, comme requis par le contexte d'application;
- d) si la détection est assurée, prendre en charge les éléments de procédure pour l'unité fonctionnelle de réponses multiples.

Annexe A

Règles d'association pour l'élément CMISE

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

A.1 Prescriptions relatives aux services ACSE, session et présentation

A.1.1 L'élément CMISE exige l'unité fonctionnelle de présentation "noyau" qui est définie dans la Rec. UIT-T X.216 | ISO/CEI 8822.

A.1.2 L'élément CMISE exige les unités fonctionnelles de session "noyau" et "transmission duplex" qui sont définies dans la Rec. UIT-T X.215 | ISO/CEI 8326.

A.1.3 L'élément CMISE exige le mode normal des services ACSE et de présentation qui sont définis dans les Rec. UIT-T X.227 | ISO/CEI 8650-1 et UIT-T X.216 | ISO/CEI 8822.

A.2 Règles d'initialisation de l'association

A.2.1 Demande

L'utilisateur du service CMISE qui lance l'établissement d'association doit fournir les "informations d'utilisateur" de la primitive A-ASSOCIATE que définit la Rec. UIT-T X.710 | ISO/CEI 9595. Les informations d'utilisateur du protocole CMIP doivent être mises à la disposition de la machine CMIPM qui doit:

- a) construire l'information CMIPUserInfo à partir des informations fournies;
- b) placer le paramètre de version de protocole dans l'information CMIPUserInfo en indiquant le bit correspondant à chaque version prise en charge;
- c) insérer CMIPUserInfo en tant qu'informations de type EXTERNAL distinctes dans le paramètre "informations d'utilisateur" de la primitive de demande A-ASSOCIATE;
- d) attendre que les informations d'utilisateur qui sont spécifiques au service CMIS soient renvoyées dans la primitive de confirmation A-ASSOCIATE.

A.2.2 Indication

Dès réception d'une primitive d'indication A-ASSOCIATE, le paramètre CMIPUserInfo doit être mis à la disposition de la machine CMIPM qui doit:

- a) vérifier qu'au moins une des versions de protocole proposées peut être prise en charge;
- b) vérifier que le paramètre de contrôle d'accès facultatif est valide;
- c) si l'une ou l'autre des vérifications échoue, l'association doit être rejetée, le motif du paramètre de défaillance présent dans la primitive de réponse A-ASSOCIATE étant mis sur "refusé par le répondeur (permanent)". L'association n'est pas établie et cette instance de la machine CMIPM doit cesser d'exister;
- d) si les vérifications précitées réussissent, les informations ci-après, si contenues dans CMIPUserInfo, doivent être communiquées à l'utilisateur du service CMISE: unités fonctionnelles prises en charge par le fournisseur du service CMISE, contrôle d'accès et informations d'utilisateur. La machine CMIPM doit attendre la réponse de l'utilisateur du service CMISE.

A.2.3 Réponse

La primitive de réponse A-ASSOCIATE qui indique «accepté» ou «rejeté» et, si elle est acceptée, qui inclut les unités fonctionnelles, les paramètres de commande d'accès et d'information de l'utilisateur, doit être mise à la disposition de la machine CMIPM qui doit:

- a) construire l'information CMIPUserInfo qui est nécessaire pour la réponse. CMIPUserInfo doit comprendre le paramètre de la version indiquant toutes les versions du protocole CMIP qui sont prises en charge;
- b) insérer l'information CMIPUserInfo en tant qu'informations de type EXTERNAL distinctes dans le paramètre «informations d'utilisateur» de la primitive de réponse A-ASSOCIATE;
- c) si la réponse d'association indique "accepté", la version de protocole agréée est la version assortie du numéro le plus élevé admis par les deux machines CMIPM. La machine CMIPM est alors prête à accepter les primitives d'indication d'élément CMISE;
- d) si la réponse d'association indique "refusé", cette instance de la machine CMIPM doit cesser d'exister.

A.2.4 Confirmation

Dès réception de la primitive de confirmation A-ASSOCIATE, le paramètre CMIPUserInfo doit être mis à la disposition de la CMIPM qui doit:

- a) si la confirmation d'association aboutit, l'association est établie et les unités fonctionnelles, les paramètres de contrôle d'accès et d'informations d'utilisateur, s'ils sont présents dans la confirmation, sont mis à la disposition du demandeur de l'association. Les unités fonctionnelles sont censées correspondre à celles qui ont reçu le soutien de chacun des utilisateurs du service CMISE et la version de protocole est celle qui est assortie du numéro le plus élevé admis par les deux CMIPM;
- b) si la confirmation d'association échoue, l'association n'est pas établie et cette instance de la machine CMIPM doit cesser d'exister.

A.3 Règles de terminaison d'association

Chacun des utilisateurs du service CMISE peut lancer une terminaison d'association.

A.3.1 Demande

Dès réception d'une demande de terminaison d'association, les paramètres nécessaires de la primitive A-RELEASE doivent être mis à la disposition de la machine CMIPM qui doit cesser d'accepter les demandes de service et attendre la confirmation de la terminaison de l'association.

A.3.2 Indication

Dès réception d'une primitive d'indication A-RELEASE, les paramètres nécessaires d'indication A-RELEASE doivent être mis à la disposition de la machine CMIPM acceptante qui doit attendre la réponse A-RELEASE.

A.3.3 Réponse

Dès réception d'une réponse de terminaison d'association en provenance de l'utilisateur du service CMISE acceptant, les paramètres nécessaires de réponse A-RELEASE doivent être mis à la disposition de la machine CMIPM acceptante. Par la suite, cette instance de la machine CMIPM doit cesser d'exister.

A.3.4 Confirmation

Dès réception d'une primitive de confirmation A-RELEASE, les paramètres nécessaires de confirmation A-RELEASE doivent être mis à la disposition de la machine CMIPM requérante. Par la suite, cette instance de la machine CMIPM doit cesser d'exister.

A.4 Règles de rupture d'association

Chacun des utilisateurs du service CMISE peut lancer une terminaison abrupte de l'association.

D'après l'information locale disponible, si les services sous-jacents ne peuvent transmettre des informations illimitées de l'utilisateur à l'aide du service A-ABORT, le paramètre CMIPAbortInfo ne peut être inclus dans les primitives du service A-ABORT.

A.4.1 Demande A-ABORT

Dès réception d'une demande de rupture d'association, les paramètres nécessaires de demande A-ABORT comprenant les informations d'utilisateur A-ABORT définies dans la Rec. UIT-T X.710 | ISO/CEI 9595 doivent être mis à la disposition de la machine CMIPM qui doit:

- a) construire CMIPAbortInfo à partir des informations fournies;
- b) régler le paramètre "source de rupture" dans CMIPUserInfo à destination de l'utilisateur du service CMISE;
- c) insérer CMIPAbortInfo comme champ distinct dans le paramètre "informations d'utilisateur" de la primitive de demande A-ABORT;
- d) par la suite, cette instance de la machine CMIPM doit cesser d'exister.

A.4.2 Indication A-ABORT

Dès réception d'une primitive d'indication A-ABORT, les paramètres nécessaires d'indication A-ABORT comprenant CMIPAbortInfo doivent être mis à la disposition de la machine CMIPM. Par la suite, cette instance de la machine CMIPM doit cesser d'exister.

A.4.3 Indication A-P-ABORT

Dès réception d'une primitive d'indication A-P-ABORT, les paramètres nécessaires d'indication A-P-ABORT doivent être mis à la disposition de la machine CMIPM. Par la suite, cette instance de la machine CMIPM doit cesser d'exister.

A.4.4 Erreur de protocole CMIP

Dès qu'elle détecte une erreur de protocole, la machine CMIPM doit:

- a) construire CMIPAbortInfo, le paramètre "source de la rupture" étant réglé sur le fournisseur du service CMISE;
- b) indiquer à l'utilisateur du service CMISE qu'une erreur de protocole s'est produite;
- c) insérer CMIPAbortInfo comme champ distinct dans le paramètre "informations d'utilisateur" de la primitive de demande A-ABORT;
- d) par la suite, cette instance de la machine CMIPM doit cesser d'exister.

Annexe B

Extension de la syntaxe ASN.1

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

La présente annexe indique de quelle façon les macros OPERATION (opération) et ERROR (erreur) de la Rec. X.219 du CCITT | ISO/CEI 9072-1 sont développées en types et sous-types de données ASN.1.

En cas d'incohérence entre ces définitions et celles de l'article 7, ces dernières l'emportent.

-- Protocole commun d'information de gestion (CMIP)

CMIP-1 {joint-iso-itu-t ms(9) cmip(1) modules(0) protocol(3)}

DEFINITIONS ::= BEGIN

-- Définitions relatives aux opérations distantes

IMPORTS OPERATION, ERROR FROM Remote-Operation-Notation {joint-iso-ccitt remote-operations(4) notation(0)}

-- Définitions du service d'annuaire

DistinguishedName, RDNSSequence

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

-- Opérations de l'élément CMISE

ROSEapdu ::= CHOICE {

 roiv-apdu [1] IMPLICIT ROIVapdu,
 rors-apdu [2] IMPLICIT RORSapdu,
 roer-apdu [3] IMPLICIT ROERapdu,
 rorj-apdu [4] IMPLICIT RORJapdu
}

ROIVapdu ::= SEQUENCE {

 invokeID InvokeIDType,
 linked-ID [0] IMPLICIT InvokeIDType OPTIONAL,
 operation-value OPERATION,
 argument ANY DEFINED BY operation-value OPTIONAL
}

RORSapdu ::= SEQUENCE {

 invokeID InvokeIDType,
 SEQUENCE { operation-value OPERATION,
 result ANY DEFINED BY operation-value } OPTIONAL
}

ROERapdu ::= SEQUENCE {

 invokeID InvokeIDType,
 error-value ERROR,
 parameter ANY DEFINED BY error-value OPTIONAL
}

RORJapdu ::= SEQUENCE {

 invokeID CHOICE {InvokeIDType,
 NULL },
 problem CHOICE { [0] IMPLICIT GeneralProblem,
 [1] IMPLICIT InvokeProblem,
 [2] IMPLICIT ReturnResultProblem,
 [3] IMPLICIT ReturnErrorProblem }
}

InvokeIDType ::= INTEGER

-- L'utilisation des codes GeneralProblem, ReturnResultProblem et ReturnErrorProblem dépend de la réalisation locale.

GeneralProblem ::= INTEGER {

 unrecognisedAPDU (0), -- détecté par le fournisseur d'élément ROSE
 mistypedAPDU (1),
 badlyStructuredAPDU (2)
}

```

InvokeProblem ::= INTEGER {
  duplicateInvocation      (0), -- détecté par l'utilisateur d'élément ROSE
  unrecognisedOperation   (1),
  mistypedArgument        (2),
  resourceLimitation      (3),
  initiatorReleasing     (4),
  unrecognisedLinkedID    (5),
  linkedResponseUnexpected (6),
  unexpectedChildOperation (7)
}

```

```

ReturnResultProblem ::= INTEGER {
  unrecognisedInvocation   (0), -- détecté par l'utilisateur d'élément ROSE
  resultResponseUnexpected (1),
  mistypedResult          (2)
}

```

```

ReturnErrorProblem ::= INTEGER {
  unrecognisedInvocation   (0), -- détecté par l'utilisateur d'élément ROSE
  errorResponseUnexpected (1),
  unrecognisedError       (2),
  unexpectedError         (3),
  mistypedParameter      (4)
}

```

-- Cette partie de la spécification ASN.1 fournit une définition du sous-type *InvokeProblem* utilisé par le protocole CMIP.

```

InvokeProblem-CMIPUser ::= InvokeProblem (
  duplicateInvocation      |
  unrecognisedOperation   |
  mistypedArgument        |
  resourceLimitation      |
)

```

-- Cette partie de la spécification ASN.1 fournit une définition des sous-types d'apdu ROIV et d'apdu RORS utilisés par le protocole CMIP. Les sous-types de l'apdu ROIV définissent les valeurs autorisées de la valeur de l'opération et de l'argument définies par cette valeur de l'opération pour toutes les notifications et opérations CMIP. Les sous-types de l'apdu RORS définissent les valeurs autorisées de la valeur de l'opération et du résultat définies par cette valeur de l'opération pour toutes les notifications et opérations du protocole CMIP

m-Action OPERATION ::= localValue : 6

```

ROIV-m-Action ::= ROIVapdu (WITH COMPONENTS {
  invokeID      PRESENT,
  linked-ID     ABSENT,
  operation-value (m-Action),
  argument      (INCLUDES ActionArgument) } )

```

m-Action-Confirmed OPERATION ::= localValue : 7

```

ROIV-m-Action-Confirmed ::= ROIVapdu (WITH COMPONENTS {
  invokeID      PRESENT,
  linked-ID     ABSENT,
  operation-value (m-Action-Confirmed),
  argument      (INCLUDES ActionArgument) } )

```

```

RORS-m-Action-Confirmed ::= RORSapdu (WITH COMPONENTS {
  ... ,
  invokeID      PRESENT,
  -- séquence de résultat -- (WITH COMPONENTS
  { operation-value (m-Action-Confirmed),
    result          (INCLUDES ActionResult) } )
  -- requis uniquement s'il existe une seule réponse à l'apdu ROIV d'action de gestion ROIV de type confirmé
  -- (ROIV-m-Action Confirmed ROIVapdu) et les données doivent être renvoyées dans l'apdu RORS
)

```

m-Cancel-Get OPERATION ::= localValue : 10

ROIV-m-Cancel-Get ::= ROIVapdu (WITH COMPONENTS {
invokeID PRESENT,
linked-ID ABSENT,
operation-value (m-Cancel-Get),
argument (INCLUDES InvokeIDType)
})

RORS-m-Cancel-Get ::= RORSapdu (WITH COMPONENTS {
invokeID PRESENT,
-- il n'y a pas de séquence de résultat pour l'opération RORS-m-Cancel-Get
})

m-Create OPERATION ::= localValue : 8

ROIV-m-Create ::= ROIVapdu (WITH COMPONENTS {
invokeID PRESENT,
linked-ID ABSENT,
operation-value (m-Create),
argument (INCLUDES CreateArgument) }
})

RORS-m-Create ::= RORSapdu (WITH COMPONENTS {
... ,
invokeID PRESENT,
-- séquence de résultat -- (WITH COMPONENTS
{ operation-value (m-Create),
result (INCLUDES CreateResult) }
})

m-Delete OPERATION ::= localValue : 9

ROIV-m-Delete ::= ROIVapdu (WITH COMPONENTS {
invokeID PRESENT,
linked-ID ABSENT,
operation-value (m-Delete),
argument (INCLUDES DeleteArgument) }
})

RORS-m-Delete ::= RORSapdu (WITH COMPONENTS {
... ,
invokeID PRESENT,
-- séquence de résultat -- (WITH COMPONENTS
{ operation-value (m-Delete),
result (INCLUDES DeleteResult) }
-- requis uniquement s'il existe une seule réponse à l'apdu ROIV de suppression m-ROIV (ROIV-m-DeleteROIVapdu)
-- et les données doivent être renvoyées dans l'apdu RORS
})

m-EventReport OPERATION ::= localValue : 0

ROIV-m-EventReport ::= ROIVapdu (WITH COMPONENTS {
invokeID PRESENT,
linked-ID ABSENT,
operation-value (m-EventReport),
argument (INCLUDES EventReportArgument) }
})

m-EventReport-Confirmed OPERATION ::= localValue : 1

ROIV-m-EventReport-Confirmed ::= ROIVapdu (WITH COMPONENTS {
invokeID PRESENT,
linked-ID ABSENT,
operation-value (m-EventReport-Confirmed),
argument (INCLUDES EventReportArgument) }
})

RORS-m-EventReport-Confirmed ::= RORSapdu (WITH COMPONENTS {
... ,
invokeID PRESENT,
-- séquence de résultat -- (WITH COMPONENTS
{ operation-value (m-EventReport-Confirmed),
result (INCLUDES EventReportResult) }
-- requis uniquement si les données doivent être renvoyées dans l'apdu RORS
})

m-Get OPERATION ::= localValue : 3

ROIV-m-Get ::= ROIVapdu (WITH COMPONENTS {
 invokeID PRESENT,
 linked-ID ABSENT,
 operation-value (m-Get),
 argument (INCLUDES GetArgument) })

RORS-m-Get ::= RORSapdu (WITH COMPONENTS {
 ... ,
 invokeID PRESENT,
 -- séquence de résultat -- (WITH COMPONENTS
 { operation-value (m-Get),
 result (INCLUDES GetResult) })
 -- requis uniquement s'il existe une seule réponse à l'apdu ROIV-m-Get
 })

m-Linked-Reply OPERATION ::= localValue : 2

ROIV-m-Linked-Reply ::= ROIVapdu (WITH COMPONENTS {
 invokeID PRESENT,
 linked-ID PRESENT,
 operation-value (m-Linked-Reply),
 argument (INCLUDES LinkedReplyArgument) })

-- Cette partie de la spécification ASN.1 fournit une définition des sous-types de réponse liée m-ROIV
 -- (ROIV-m-Linked-Reply) utilisés par le protocole CMIP. Les sous-types de l'apdu ROIV de réponse liée m-ROIV
 -- (ROIV-m-Linked-Reply ROIVapdu) indiquent les valeurs autorisées de l'argument définies par la valeur de
 -- l'opération pour les opérations spécifiques de réponse liée CMIP

ROIV-m-Linked-Reply-Action ::= ROIV-m-Linked-Reply (WITH COMPONENTS {
 invokeID PRESENT,
 linked-ID PRESENT,
 operation-value (m-Linked-Reply),
 argument (INCLUDES LinkedReplyArgument (WITH COMPONENTS {
 getResult ABSENT,
 getListError ABSENT,
 setResult ABSENT,
 setListError ABSENT,
 actionResult PRESENT,
 processingFailure PRESENT,
 deleteResult ABSENT,
 actionError PRESENT,
 deleteError ABSENT })
) })

ROIV-m-Linked-Reply-Delete ::= ROIV-m-Linked-Reply (WITH COMPONENTS {
 invokeID PRESENT,
 linked-ID PRESENT,
 operation-value (m-Linked-Reply),
 argument (INCLUDES LinkedReplyArgument (WITH COMPONENTS {
 getResult ABSENT,
 getListError ABSENT,
 setResult ABSENT,
 setListError ABSENT,
 actionResult ABSENT,
 processingFailure PRESENT,
 deleteResult PRESENT,
 actionError ABSENT,
 deleteError PRESENT })
) })

ROIV-m-Linked-Reply-Get ::= ROIV-m-Linked-Reply (WITH COMPONENTS {
 invokeID PRESENT,
 linked-ID PRESENT,
 operation-value (m-Linked-Reply),
 argument (INCLUDES LinkedReplyArgument (WITH COMPONENTS {
 getResult PRESENT,
 getListError PRESENT,
 setResult ABSENT,
 setListError ABSENT,
 actionResult ABSENT,
 processingFailure ABSENT,
 deleteResult ABSENT,
 actionError ABSENT,
 deleteError ABSENT })
) })

```

        actionResult      ABSENT,
        processingFailure  PRESENT,
        deleteResult      ABSENT,
        actionError       ABSENT,
        deleteError       ABSENT } )

```

```

    ) ) )

```

```

ROIV-m-Linked-Reply-Set ::= ROIV-m-Linked-Reply (WITH COMPONENTS {
  invokeID      PRESENT,
  linked-ID     PRESENT,
  operation-value (m-Linked-Reply),
  argument      (INCLUDES LinkedReplyArgument (WITH COMPONENTS {
    getResult      ABSENT,
    getListError   ABSENT,
    setResult      PRESENT,
    setListError   PRESENT,
    actionResult   ABSENT,
    processingFailure PRESENT,
    deleteResult   ABSENT,
    actionError    ABSENT,
    deleteError    ABSENT } )
  ) ) )

```

```

    ) ) )

```

```

m-Set OPERATION ::= localValue : 4

```

```

ROIV-m-Set ::= ROIVapdu (WITH COMPONENTS {
  invokeID      PRESENT,
  linked-ID     ABSENT,
  operation-value (m-Set),
  argument      (INCLUDES SetArgument) } )

```

```

m-Set-Confirmed OPERATION ::= localValue : 5

```

```

ROIV-m-Set-Confirmed ::= ROIVapdu (WITH COMPONENTS {
  invokeID      PRESENT,
  linked-ID     ABSENT,
  operation-value (m-Set-Confirmed),
  argument      (INCLUDES SetArgument) } )

```

```

RORS-m-Set-Confirmed ::= RORSapdu (WITH COMPONENTS {

```

```

  ... ,
  invokeID      PRESENT,
  -- séquence de résultat -- (WITH COMPONENTS
  { operation-value (m-Set-Confirmed),
    result          (INCLUDES SetResult) } )
  -- requis uniquement s'il existe une seule réponse à l'apdu ROIV d'affectation de gestion ROIV en mode confirmé
  -- (ROIV-m-Set-Confirmed ROIVapdu)
  } )

```

```

-- Cette partie de la spécification ASN.1 fournit une définition des sous-types d'apdu ROER utilisés par le
-- protocole CMIP. Les sous-types de l'apdu ROER indiquent les valeurs autorisées de la valeur d'erreur et du
-- paramètre définies par cette valeur d'erreur pour toutes les notifications et opérations CMIP

```

```

accessDenied ERROR ::= localValue : 2

```

```

ROER-accessDenied ::= ROERapdu (WITH COMPONENTS {
  invokeID      PRESENT,
  error-value   (accessDenied) } )

```

```

-- Cette apdu ROER peut seulement être renvoyée en réponse aux apdu ROIV-m-Get, ROIV-m-Set-Confirmed,
-- ROIV-m-Action-Confirmed, ROIV-m-Create et ROIV-m-Delete

```

```

classInstanceConflict ERROR ::= localValue : 19

```

```

ROER-classInstanceConflict ::= ROERapdu (WITH COMPONENTS {
  invokeID      PRESENT,
  error-value   (classInstanceConflict),
  parameter     (INCLUDES BaseManagedObjectId) } )

```

```

-- Cette apdu ROER peut seulement être renvoyée en réponse aux apdu ROIV-m-Get, ROIV-m-Set-Confirmed,
-- ROIV-m-Action-Confirmed, ROIV-m-Create et ROIV-m-Delete

```

complexityLimitation ERROR ::= localValue : 20

ROER-complexityLimitation ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (complexityLimitation),
parameter (INCLUDES ComplexityLimitation) OPTIONAL })

-- Cette apdu ROER peut seulement être renvoyée en réponse aux apdu ROIV-m-Get, ROIV-m-Set-Confirmed,
 -- ROIV-m-Action-Confirmed, ROIV-m-Create et ROIV-m-Delete

duplicateManagedObjectInstance ERROR ::= localValue : 11

ROER-duplicateManagedObjectInstance::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (duplicateManagedObjectInstance),
parameter (INCLUDES ObjectInstance) })

-- Cette apdu ROER peut seulement être renvoyée en réponse à l'apdu ROIV-m-Create)

getListError ERROR ::= localValue : 7

ROER-getListError ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (getListError),
parameter (INCLUDES GetListError) })

-- Cette apdu ROER peut seulement être renvoyée en réponse à l'apdu ROIV-m-Get

invalidArgumentValue ERROR ::= localValue : 15

ROER-invalidArgumentValue ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (invalidArgumentValue),
parameter (INCLUDES InvalidArgumentValue) })

-- Cette apdu ROER peut seulement être renvoyée en réponse aux apdu ROIV-m-EventReport-Confirmed et
 -- ROIV-m-Action-Confirmed)

invalidAttributeValue ERROR ::= localValue : 6

ROER-invalidAttributeValue ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (invalidAttributeValue),
parameter (INCLUDES Attribute) })

-- Cette apdu ROER peut seulement être renvoyée en réponse à l'apdu ROIV-m-Create

invalidFilter ERROR ::= localValue : 4

ROER-invalidFilter ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (invalidFilter),
parameter (INCLUDES CMISFilter) })

-- Cette apdu ROER peut seulement être renvoyée en réponse aux apdu ROIV-m-Get, ROIV-m-Set-Confirmed,
 -- ROIV-m-Action-Confirmed et ROIV-m-Delete

invalidObjectInstance ERROR ::= localValue : 17

ROER-invalidObjectInstance ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (invalidObjectInstance),
parameter (INCLUDES ObjectInstance) })

-- Cette apdu ROER peut seulement être renvoyée en réponse à l'apdu ROIV-m-Create

invalidScope ERROR ::= localValue : 16

ROER-invalidScope ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (invalidScope),
parameter (INCLUDES Scope) })

-- Cette apdu ROER peut seulement être renvoyée en réponse aux apdu ROIV-m-Get, ROIV-m-Set-Confirmed,
 -- ROIV-m-Action-Confirmed et ROIV-m-Delete

missingAttributeValue ERROR ::= localValue : 18

**ROER-missingAttributeValue ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (missingAttributeValue),
 parameter (INCLUDES SET OF AttributeId) })**

-- Cette apdu ROER peut seulement être renvoyée en réponse à l'apdu ROIV-m-Create

mistypedOperation ERROR ::= localValue : 21

**ROER-mistypedOperation ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (mistypedOperation) })**

-- Cette apdu ROER peut seulement être renvoyée en réponse à l'apdu ROIV m-Cancel-Get

noSuchAction ERROR ::= localValue : 9

**ROER-noSuchAction ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (noSuchAction),
 parameter (INCLUDES NoSuchAction) })**

-- Cette apdu ROER peut seulement être renvoyée en réponse à l'apdu ROIV-m-Action-Confirmed

noSuchArgument ERROR ::= localValue : 14

**ROER-noSuchArgument ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (noSuchArgument),
 parameter (INCLUDES NoSuchArgument) })**

-- Cette apdu ROER peut seulement être renvoyée en réponse aux apdu ROIV-m-EventReport-Confirmed et

-- ROIV-m-Action-Confirmed

noSuchAttribute ERROR ::= localValue : 5

**ROER-noSuchAttribute ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (noSuchAttribute),
 parameter (INCLUDES AttributeId) })**

-- Cette apdu ROER peut seulement être renvoyée en réponse à l'apdu ROIV-m-Create

noSuchEventType ERROR ::= localValue : 13

**ROER-noSuchEventType ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (noSuchEventType),
 parameter (INCLUDES NoSuchEventType) })**

-- Cette apdu ROER peut seulement être renvoyée en réponse à l'apdu ROIV-m-EventReport-Confirmed

noSuchInvokeId ERROR ::= localValue : 22

**ROER-noSuchInvokeId ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (noSuchInvokeId),
 parameter (INCLUDES InvokeIDType) })**

-- Cette apdu ROER peut seulement être renvoyée en réponse à l'apdu ROIV-m-Cancel-Get

noSuchObjectClass ERROR ::= localValue : 0

**ROER-noSuchObjectClass ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (noSuchObjectClass),
 parameter (INCLUDES ObjectClass) })**

-- Cette apdu ROER peut seulement être renvoyée en réponse aux apdu ROIV-m-EventReport-Confirmed,

-- ROIV-m-Get, ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed, ROIV-m-Create et ROIV-m-Delete

noSuchObjectInstance ERROR ::= localValue : 1

ROER-noSuchObjectInstance ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (noSuchObjectInstance),
parameter (INCLUDES ObjectInstance) })

-- Cette apdu ROER peut seulement être renvoyée en réponse aux apdu ROIV-m-EventReport-Confirmed,
 -- ROIV-m-Get, ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed, ROIV-m-Create et ROIV-m-Delete

noSuchReferenceObject ERROR ::= localValue : 12

ROER-noSuchReferenceObject ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (noSuchReferenceObject),
parameter (INCLUDES ObjectInstance) })

-- Cette apdu ROER peut seulement être renvoyée en réponse à l'apdu ROIV-m-Create

operationCancelled ERROR ::= localValue : 23

ROER-operationCancelled ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (operationCancelled) })

-- Cette apdu ROER peut seulement être renvoyée en réponse à l'apdu ROIV-m-Get

processingFailure ERROR ::= localValue : 10

ROER-processingFailure ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (processingFailure),
parameter (INCLUDES ProcessingFailure) OPTIONAL })

-- Cette apdu ROER peut seulement être renvoyée en réponse aux apdu ROIV-m-EventReport-Confirmed,
 -- ROIV-m-Get, ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed, ROIV-m-Create et ROIV-m-Delete

setListError ERROR ::= localValue : 8

ROER-setListError ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (setListError),
parameter (INCLUDES SetListError) })

-- Cette apdu ROER peut seulement être renvoyée en réponse à l'apdu ROIV-m-Set-Confirmed

syncNotSupported ERROR ::= localValue : 3

ROER-syncNotSupported ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (syncNotSupported),
parameter (INCLUDES CMISync) })

-- Cette apdu ROER peut seulement être renvoyée en réponse aux apdu ROIV-m-Get, ROIV-m-Set-Confirmed,
 -- ROIV-m-Action-Confirmed, et ROIV-m-Delete

-- Pour compléter la spécification de la syntaxe abstraite fournie dans cette annexe, on pourra se référer aux
 -- définitions des types supports (7.4)

END -- FIN des définitions de la syntaxe du protocole CMIP

Annexe C

Exemples de données APDU d'élément ROSE du service CMISE

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

La présente annexe fournit quelques exemples du développement complet des unités APDU d'élément ROSE acheminant l'information de protocole CMIP.

Ces exemples sont fournis pour aider les utilisateurs de la présente Recommandation | Norme internationale.

-- apdu ROIV pour l'opération d'action CMISE de type confirmé

```
ROIVapdu-example ::= [1] IMPLICIT SEQUENCE {
  invokeID          InvokeIDType,
  operation-value   INTEGER {m-Action-Confirmed (7)},
  argument          SEQUENCE {
    COMPONENTS OF
      accessControl [5] AccessControl OPTIONAL,
      synchronization [6] IMPLICIT CMISSync OPTIONAL,
      scope          [7] Scope DEFAULT baseObject,
      filter         CMISFilter DEFAULT and {},
  actionInfo       [12] IMPLICIT SEQUENCE {
  actionType       ActionTypeId,
  actionInfoArg    [4] ANY DEFINED BY actionType OPTIONAL
  } } }
```

-- apdu RORS pour l'opération d'action CMISE de type confirmé

```
RORSapdu-example ::= [2] IMPLICIT SEQUENCE {
  invokeID          InvokeIDType,
  SEQUENCE {
    operation-value INTEGER {m-Action-Confirmed (7)},
    result          SEQUENCE {
      managedObjectClass ObjectClass OPTIONAL,
      managedObjectInstance ObjectInstance OPTIONAL,
      currentTime        [5] IMPLICIT GeneralizedTime OPTIONAL,
      actionReply        [6] IMPLICIT SEQUENCE {
      actionType         ActionTypeId,
      actionReplyInfo    [4] ANY DEFINED BY actionType } OPTIONAL
    } } }
```

-- apdu ROIV pour la réponse liée CMISE pour une opération d'action de type confirmé

```
ROIVapdu-linked-example ::= [1] IMPLICIT SEQUENCE {
  invokeID          InvokeIDType,
  linked-ID         [0] IMPLICIT InvokeIDType,
  operation-value   INTEGER {m-Action-Confirmed (7)},
  argument          CHOICE {
    actionResult      [4] IMPLICIT ActionResult,
    processingFailure [5] IMPLICIT ProcessingFailure,
    actionError       [7] IMPLICIT ActionError
  } } }
```

-- apdu ROER pour l'opération d'action CMISE de type confirmé lorsqu'une erreur noSuchAction se produit

```
ROERapdu-example ::= [3] IMPLICIT SEQUENCE {
  invokeID          InvokeIDType,
  error-value       INTEGER {noSuchAction (9)},
  parameter         SEQUENCE { managedObjectClass ObjectClass OPTIONAL,
                                actionId           ActionTypeId
  } } }
```

Annexe D**Syntaxe abstraite d'annuaire**

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

La présente Recommandation | Norme internationale reprend des spécifications de la première édition de l'annuaire. La présente annexe contient un extrait de la Rec. X.501 du CCITT (1988) | ISO/CEI 9594-2:1990 afin qu'une copie de la syntaxe nécessaire figure dans une publication en vigueur.

La norme relative à l'annuaire a des règles d'extensibilité (spécifiées dans la Rec. UIT-T X.519 (1993) | ISO/CEI 9594-5:1995) stipulant que si un système d'annuaire de 1990 est confronté à de la syntaxe étendue, il doit ignorer ce qu'il ne comprend pas. Cela pourrait avoir des répercussions sur les corrélations de noms pour des objets gérés de l'OSI comportant des entrées d'annuaire telles qu'un parent de dénomination, étant donné que les composantes de nom distinctif relatif de l'annuaire peuvent utiliser la syntaxe étendue. Dans un tel cas, des composantes facultatives de la syntaxe étendue des noms distinctifs (DistinguishedName) devront être supprimées avant que soit formé, pour l'objet géré ayant comme parent de dénomination une telle entrée d'annuaire, le nom distinctif de gestion OSI.

Les productions suivantes, utilisées pour la syntaxe des noms distinctifs importée par le protocole CMIP et par les autres normes, sont reproduites dans la présente annexe à partir du module ASN.1,

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

initialement publié dans la Rec. X.501 du CCITT (1988) | ISO/CEI 9594-2:1990.

AttributeType ::= OBJECT IDENTIFIER

AttributeValue ::= ANY

AttributeValueAssertion ::= SEQUENCE {
 type AttributeType,
 assertion AttributeValue}

Name ::= CHOICE { rdnSequence RDNSequence }

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName

DistinguishedName ::= RDNSequence

RelativeDistinguishedName ::= SET OF AttributeValueAssertion

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	RGT et maintenance des réseaux: 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 Y	Infrastructure mondiale de l'information
Série Z	Langages de programmation