

**Remplacée par une version plus récente**



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

**CCITT**

**X.711**

COMITÉ CONSULTATIF  
INTERNATIONAL  
TÉLÉGRAPHIQUE ET TÉLÉPHONIQUE

(03/91)

**RÉSEAUX DE COMMUNICATIONS DE DONNÉES:  
INTERCONNEXION DE SYSTÈMES OUVERTS (OSI);  
GESTION**

---

**SPÉCIFICATION DU PROTOCOLE COMMUN  
DE TRANSFERT D'INFORMATIONS  
DE GESTION POUR LES APPLICATIONS  
DU CCITT**

**Recommandation X.711**  
Remplacée par une version plus récente

---



Genève, 1991

# Remplacée par une version plus récente

## AVANT-PROPOS

Le CCITT (Comité consultatif international télégraphique et téléphonique) est un organe permanent de l'Union internationale des télécommunications (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.

L'Assemblée plénière du CCITT, qui se réunit tous les quatre ans, détermine les thèmes d'études et approuve les Recommandations rédigées par ses Commissions d'études. Entre les Assemblées plénières, l'approbation des Recommandations par les membres du CCITT s'effectue selon la procédure définie dans la Résolution n° 2 du CCITT (Melbourne, 1988).

La Recommandation X.711, que l'on doit à la Commission d'études VII, a été approuvée le 22 mars 1991 selon la procédure définie dans la Résolution n° 2.

---

## NOTE DU CCITT

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

© UIT 1991

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

# Remplacée par une version plus récente

## Recommandation X.711

### SPÉCIFICATION DU PROTOCOLE COMMUN DE TRANSFERT D'INFORMATIONS DE GESTION POUR LES APPLICATIONS DU CCITT<sup>1)</sup>

#### SOMMAIRE

- 1 *Domaine d'application*
- 2 *Références*
- 3 *Définitions*
- 4 *Symboles et abréviations*
- 5 *Présentation générale*
- 6 *Eléments de procédure*
- 7 *Syntaxe abstraite*
- 8 *Conformité*

*Annexe A* – Règles d'association pour le CMISE

*Annexe B* – Extension de la syntaxe ASN.1

*Annexe C* – Exemples d'APDU ROSE du CMISE

## 1 **Domaine d'application**

La présente Recommandation spécifie un protocole qui est utilisé par des entités de la couche application pour échanger des informations de gestion.

La présente Recommandation spécifie:

- les procédures de transmission des informations de gestion entre entités d'application;
- la syntaxe abstraite du protocole commun de transfert d'informations (CMIP) de gestion 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.

La présente Recommandation ne spécifie pas:

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

## 2 **Références**

- [1] Recommandation du CCITT *Modèle de référence pour l'interconnexion de systèmes ouverts pour les applications du CCITT*, Livre bleu, Fascicule VIII.4, Rec. X.200, UIT, Genève, 1988.
- [2] Norme ISO/CEI 7498-4 *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base – Partie 4: Cadre général de gestion*, 1989.

---

<sup>1)</sup> La Recommandation X.711 et la norme ISO/CEI 9596, *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Protocole commun de transfert d'informations de gestion, Partie 1 – Spécification*, ont été élaborées en étroite coopération et sont alignées du point de vue technique.

# Remplacée par une version plus récente

- [3] Recommandation du CCITT *Spécification de la syntaxe abstraite numéro un (ASN.1)*, Livre bleu, Fascicule VIII.4, Rec. X.208, UIT, Genève, 1988.
- [4] Recommandation du CCITT *Spécification des règles de codage pour la notation de syntaxe abstraite numéro un (ASN.1)*, Livre bleu, Fascicule VIII.4, Rec. X.209, UIT, Genève, 1988.
- [5] Recommandation du CCITT *Définition du service de session pour l'interconnexion de systèmes ouverts (OSI) pour les applications du CCITT*, Livre bleu, Fascicule VIII.4, Rec. X.215, UIT, Genève, 1988.
- [6] Recommandation du CCITT *Définition du service de présentation de l'OSI (interconnexion de systèmes ouverts) pour les applications du CCITT*, Livre bleu, Fascicule VIII.4, Rec. X.216, UIT, Genève, 1988.
- [7] Recommandation du CCITT *Définition du service de contrôle d'association pour l'interconnexion de systèmes ouverts pour les applications du CCITT*, Livre bleu, Fascicule VIII.4, Rec. X.217, UIT, Genève, 1988.
- [8] Recommandation du CCITT *Téléopérations: modèle, notation et définition du service*, Livre bleu, Fascicule VIII.4, Rec. X.219, UIT, Genève, 1988.
- [9] Recommandation du CCITT *Spécification du protocole de présentation de l'OSI (interconnexion de systèmes ouverts) pour les applications du CCITT*, Livre bleu, Fascicule VIII.5, Rec. X.226, UIT, Genève, 1988.
- [10] Recommandation du CCITT *Spécification du service de contrôle d'association de l'OSI (interconnexion de systèmes ouverts) pour les applications du CCITT*, Livre bleu, Fascicule VIII.5, Rec. X.227, UIT, Genève, 1988.
- [11] Recommandation du CCITT *Opérations distantes: spécification du protocole*, Livre bleu, Fascicule VIII.5, Rec. X.229, UIT, Genève, 1988.
- [12] Recommandation du CCITT *Définition du service commun de transfert d'informations de gestion pour les applications du CCITT*, Rec. X.710, UIT, Genève, 1991.

## 3 Définitions

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

### 3.1 Définitions du modèle de référence de base

La présente Recommandation utilise les termes ci-après, définis dans la Recommandation X.200 [1]:

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

### 3.2 Définitions du cadre général de gestion

La présente Recommandation utilise les termes ci-après, définis dans la norme ISO/CEI 7498-4 [2]:

- a) objet géré;
- b) information de gestion;

# Remplacée par une version plus récente

- c) base d'information de gestion;
- d) entité d'application de gestion de systèmes.

## 3.3 *Définitions relatives aux opérations distantes*

La présente Recommandation utilise les termes ci-après, définis dans la Recommandation X.219 [8]:

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

## 3.4 *Définitions du CMIS*

La présente Recommandation utilise les termes ci-après, définis dans la Recommandation X.710 [12]:

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

## 3.5 *Définitions de l'ACSE*

La présente Recommandation utilise les termes ci-après définis dans la Recommandation X.217 [7]:

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

## 3.6 *Définitions relatives à la présentation*

La présente Recommandation utilise les termes ci-après définis dans la Recommandation X.216 [6]:

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

## 4 **Symboles et abréviations**

ACSE	Élément de service de contrôle d'association (Association Control Service Element)
APDU	Unité de données du protocole d'application (Application Protocol Data Unit)
ASE	Élément de service d'application (Application Service Element)
ASN.1	Notation de syntaxe abstraite numéro un (Abstract Syntax Notation One)

# Remplacée par une version plus récente

CMIP	Protocole commun d'informations de gestion (Common Management Information Protocol)
CMIPM	Machine du protocole commun de transfert d'informations de gestion (Common Management Information Protocol Machine)
CMIS	Service commun de transfert d'informations de gestion (Common Management Information service)
CMISE	Élément de service commun de transfert d'informations de gestion (Common Management Information Service Element)
DCS	Ensemble des contextes définis (defined context set)
PCI	Information de contrôle du protocole (protocol control information)
PDU	Unité de données du protocole (protocol data unit)
PICS	Déclaration de conformité d'une mise en œuvre de protocole (protocol implementation conformance statement)
RO	Opérations distantes (remote operations)
ROSE	Élément de service d'opérations distantes (remote operations service element)
SMAE	Entité d'application de gestion de systèmes (systems management application entity).

## 5 Présentation générale

Le protocole commun de transfert d'informations de gestion (CMIP) spécifie des éléments de protocole qui peuvent être utilisés pour fournir les services d'opérations et de notifications décrits dans la Recommandation X.710 [12], qui définit les services communs de transfert d'informations de gestion (CMIS).

### 5.1 Services fournis

Le protocole spécifié dans la présente Recommandation assure les services définis dans la Recommandation X.710 [12]. Ces services sont indiqués dans le tableau 1/X.711

TABLEAU 1/X.711

#### Services communs de transfert d'informations de gestion

Service	Type
G-OBTENTION-ANNULATION	Confirmé
G-NOTIFICATION-ÉVÉNEMENT	Confirmé/non confirmé
G-OBTENTION	Confirmé
G-MODIFICATION	Confirmé/non confirmé
G-ACTION	Confirmé/non confirmé
G-CRÉATION	Confirmé
G-SUPPRESSION	Confirmé

# Remplacée par une version plus récente

## 5.2 Services sous-jacents

La présente Recommandation utilise les services APPEL DE RO, RÉSULTAT DE RO, ERREUR DE RO et REFUS DE RO PAR L'UTILISATEUR de l'élément de service d'opérations distantes (ROSE) qui sont définis dans la Recommandation X.219 [8]. Le ROSE suppose l'utilisation du service de présentation défini dans la Recommandation X.216 [6]. Les opérations de type «confirmé» du CMIP sont la classe d'opérations 2 (asynchrone) ou la classe d'opérations 1 (synchrone) comme le demande l'application. Le choix de la classe d'opérations doit être décidé sur le plan local. Les opérations de type «non confirmé» du CMIP sont celles de la classe d'opérations 5 (asynchrone, résultat non signalé). Le 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 du protocole d'application ROSE (ROSEapdus) peuvent être mis en correspondance avec d'autres services de présentation que le service de TRANSFERT DE DONNÉES DE PRÉSENTATION.

*Remarque* – Par exemple, il peut être nécessaire de modifier l'ensemble des contextes définis (DCS) lorsque l'opération CMIP est envoyée à l'utilisateur de service CMISE équivalent. Dans ce cas, l'APDU ROSE qui achemine l'opération CMIP sera mise en correspondance avec le service de MODIFICATION DE CONTEXTES DE PRÉSENTATION qui sert également à apporter les modifications au DCS.

Des détails sur les autres services de présentation requis et 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 suppose l'utilisation des services suivants: ÉTABLISSEMENT D'ASSOCIATION, TERMINAISON D'ASSOCIATION, RUPTURE D'ASSOCIATION ET RUPTURE D'ASSOCIATION PAR LE FOURNISSEUR, de l'élément de service de contrôle d'association.

### 5.2.2 Services supposés être fournis par la couche de présentation

La Recommandation X.229 [11] suppose l'utilisation du service de TRANSFERT DE DONNÉES DE PRÉSENTATION de la couche de présentation pour le transfert des unités de données de protocole APPEL DE RO, RÉSULTAT DE RO, ERREUR DE RO et REFUS DE RO.

## 5.3 Définitions relatives aux informations de gestion

La présente Recommandation définit la syntaxe abstraite du protocole commun de transfert d'informations de gestion. Les attributs spécifiques à un objet géré particulier sont spécifiés par la Recommandation qui définit cet objet.

## 6 Éléments de procédure

Le présent paragraphe donne la définition des éléments de procédure du CMIP. Les procédures définissent le transfert des PDU du CMIP dont la structure, le codage et la relation avec les primitives du service CMIS sont spécifiés au § 7.

La machine du protocole commun de transfert d'informations de gestion (CMIPM) reçoit des primitives de service de demande et de réponse du CMIS et envoie des PDU du CMIP en lançant des éléments de procédure spécifiques, comme cela est spécifié dans le présent paragraphe.

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

## Remplacée par une version plus récente

Les procédures indiquent uniquement comment interpréter les divers champs de la PDU du CMIP mais non ce qu'un utilisateur du service CMISE appelant ferait avec les informations qu'il demande, ni comment un utilisateur du service CMISE exécutant traiterait les informations.

### 6.1 *Etablissement d'association*

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

Un utilisateur du service CMISE peut lancer un établissement d'association à l'aide du service d'ÉTABLISSEMENT D'ASSOCIATION de la Recommandation X.217 [7].

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

### 6.2 *Opérations distantes*

#### 6.2.1 *Éléments de procédure RO*

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

- a) lancement;
- b) résultat positif;
- c) résultat négatif;
- d) refus par l'utilisateur;
- e) refus par le fournisseur.

Ces éléments de procédure sont décrits intégralement dans la Recommandation X.229 [11].

Le tableau 2/X.711 indique la correspondance entre les paramètres CMIS et ROSE.

TABLEAU 2/X.711

#### Correspondance entre les paramètres CMIS et ROSE

Paramètre CMIS	Paramètre ROSE
Identificateur d'appel	ID d'appel
Identificateur lié	ID-lié

La correspondance entre d'autres paramètres CMIS et ROSE est indiquée au § 7.

# Remplacée par une version plus récente

## 6.2.2 Paramètres du problème de REFUS DE RO

La mise en correspondance ou le traitement des paramètres du problème de REFUS DE RO sont décrits ci-après.

6.2.2.1 La mise en correspondance du problème d'appel de REFUS DE RO PAR L'UTILISATEUR avec les codes d'erreur CMISE est indiquée dans le tableau 3/X.711.

TABLEAU 3/X.711

**Mise en correspondance du problème d'appel  
de REFUS DE RO PAR L'UTILISATEUR  
avec les codes d'erreur CMISE**

Paramètre REFUS DE RO	Code d'erreur CMISE
Appel-doublé	Appel doublé
Argument-mal-typé	Argument mal typé
Limitation-de-ressources	Limitation de ressources
Opération-non-reconnue	Opération non reconnue

Les autres paramètres «problème d'appel» dépendent de la réalisation locale.

6.2.2.2 Le traitement des autres paramètres REFUS DE RO dépend de la réalisation locale.

## 6.3 Procédure de notification d'événement

### 6.3.1 Appel

Les procédures de notification d'événement sont lancées par la primitive de demande de G-NOTIFICATION-ÉVÉNEMENT.

Dès réception de la primitive de demande de G-NOTIFICATION-ÉVÉNEMENT, la CMIPM doit:

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

### 6.3.2 Réception

Dès réception d'une APDU demandant une opération de notification d'événement m ou de notification d'événement de gestion de type confirmé, la CMIPM doit, si l'APDU est bien formée, envoyer une primitive d'indication de G-NOTIFICATION-ÉVÉNEMENT à l'utilisateur du service CMISE dont le paramètre de mode indique si la confirmation est demandée ou non ou, sinon, construire une APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure de REFUS DE RO PAR L'UTILISATEUR.

# Remplacée par une version plus récente

## 6.3.3 Réponse

Dans le mode confirmé, la CMIPM doit accepter une primitive de réponse de G-NOTIFICATION-ÉVÉNEMENT et doit:

- a) construire une APDU confirmant la notification de G-NOTIFICATION-ÉVÉNEMENT;
- b) si les paramètres présents dans la primitive de réponse de G-NOTIFICATION-ÉVÉNEMENT indiquent que la notification a été acceptée, envoyer l'APDU en utilisant la procédure de RÉSULTAT de RO, sinon envoyer l'APDU en utilisant la procédure d'ERREUR DE RO.

## 6.3.4 Réception d'une réponse

Dès réception d'une APDU répondant à une notification de G-NOTIFICATION-ÉVÉNEMENT, la CMIPM doit, si l'APDU est bien formée, envoyer une primitive de confirmation de G-NOTIFICATION-ÉVÉNEMENT à l'utilisateur du service CMISE, terminant ainsi la procédure de notification, ou sinon, construire une APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure de REFUS DE RO PAR L'UTILISATEUR.

## 6.4 Procédure d'obtention

### 6.4.1 Appel

Les procédures d'obtention sont lancées par la primitive de demande de G-OBTENTION.

Dès réception de la primitive de demande de G-OBTENTION, la CMIPM doit:

- a) construire une APDU demandant l'opération de G-OBTENTION (m-Get opération);
- b) envoyer l'APDU en utilisant la procédure d'APPEL DE RO.

### 6.4.2 Réception

Dès réception d'une APDU demandant l'opération de G-OBTENTION, la CMIPM doit, si l'APDU est bien formée, envoyer une primitive d'indication de G-OBTENTION à l'utilisateur du service CMISE ou, sinon, construire une APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure REFUS DE RO PAR L'UTILISATEUR.

### 6.4.3 Réponse

La CMIPM doit:

- a) accepter aucune, une ou plusieurs primitives de réponse à une demande de G-OBTENTION contenant un indicateur lié (linked-ID), suivies d'une seule primitive de réponse à une demande de G-OBTENTION sans indicateur lié;
- b) pour chaque primitive de réponse à une demande de G-OBTENTION contenant un indicateur lié, la CMIPM doit:
  - construire une 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 demande d'obtention de la liste d'erreurs (getListError), demande d'obtention du résultat (getResult) ou défaillance du traitement (processingFailure);
  - envoyer chaque APDU en utilisant la procédure d'APPEL DE RO;
- c) pour la primitive de réponse à une demande de G-OBTENTION ne contenant pas d'indicateur lié, la CMIPM doit:
  - construire une APDU confirmant l'opération de demande de G-OBTENTION;
  - si les paramètres présents dans la primitive de réponse à une demande de G-OBTENTION indiquent que l'opération a été exécutée correctement, envoyer l'APDU en utilisant la procédure de RÉSULTAT DE RO. Si les paramètres présents dans la primitive de réponse à une demande de G-OBTENTION 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'APDU en utilisant la procédure d'ERREUR DE RO.

# Remplacée par une version plus récente

## 6.4.4. Réception de la réponse

Dès réception d'une APDU répondant à une opération G-OBTENTION (mGet), la CMIPM doit:

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

## 6.4.5 Procédure d'obtention d'annulation

### 6.4.5.1 Appel

Les procédures d'obtention d'annulation sont lancées par la primitive de demande de G-OBTENTION-ANNULATION (m-Cancel-Get).

Dès réception de la primitive de demande de G-OBTENTION-ANNULATION, la CMIPM doit:

- a) construire une APDU demandant l'opération de demande d'annulation de gestion (mCancel-Get);
- b) envoyer l'APDU en utilisant la procédure d'APPEL DE RO.

### 6.4.5.2 Réception

Dès réception d'une APDU demandant l'opération d'obtention d'annulation de gestion, la CMIPM doit, si l'APDU est bien formée, envoyer une primitive d'indication de G-OBTENTION-ANNULATION à l'utilisateur du service CMISE ou, sinon, construire une APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure de REFUS DE RO PAR L'UTILISATEUR.

### 6.4.5.3 Réponse

La CMIPM doit:

- a) construire une APDU confirmant l'opération d'obtention d'annulation de gestion;
- b) si les paramètres contenus dans la primitive de réponse de G-OBTENTION-ANNULATION indiquent que l'opération a été exécutée correctement, envoyer l'APDU en faisant appel à la procédure ERREUR DE RO. Si l'opération de G-OBTENTION-ANNULATION est bien exécutée, l'utilisateur du service CMISE exécuteur doit cesser d'envoyer des réponses liées vers l'opération de G-OBTENTION et doit envoyer une primitive de réponse de G-OBTENTION contenant l'erreur «opération annulée».

### 6.4.5.4 Réception de la réponse

Dès réception d'une APDU répondant à une opération d'obtention d'annulation de gestion, la CMIPM doit, si l'APDU est bien formée, envoyer une primitive de confirmation de G-OBTENTION-ANNULATION à l'utilisateur du service CMISE ou, sinon, construire une APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure de REFUS DE RO PAR L'UTILISATEUR.

## 6.5 Procédure de modification

### 6.5.1 Appel

Les procédures de modification sont lancées par la primitive de demande de G-MODIFICATION.

Dès réception de la primitive de demande de G-MODIFICATION, la CMIPM doit:

## Remplacée par une version plus récente

- a) dans le mode confirmé, construire une APDU demandant l'opération de modification de gestion de type confirmé (mSetConfirmed), ou sinon, construire une APDU demandant l'opération de modification de gestion (mSet);
- b) envoyer l'APDU en utilisant la procédure d'APPEL DE RO.

### 6.5.2 Réception

Dès réception d'une APDU demandant l'opération de modification de gestion ou de modification de gestion de type confirmé, la CMIPM doit, si l'APDU est bien formée, envoyer une primitive d'indication de G-MODIFICATION à l'utilisateur du service CMISE, dont le paramètre de mode indique si une confirmation est demandée ou non ou, sinon, construire une APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure de REFUS DE RO PAR L'UTILISATEUR.

### 6.5.3 Réponse

Dans le mode confirmé, la CMIPM doit:

- a) accepter aucune, une ou plusieurs primitives de réponse de G-MODIFICATION contenant un indicateur lié, suivies d'une seule primitive de réponse de G-MODIFICATION sans indicateur lié;
- b) pour chaque primitive de réponse de G-MODIFICATION contenant un identificateur lié, la CMIPM doit:
  - construire une APDU demandant l'opération de réponse liée de gestion dont l'argument de réponse liée est mis comme il se doit sur modification de la liste d'erreurs (setListError), modification du résultat (setResult) ou défaillance du traitement (processingFailure);
  - envoyer chaque APDU en utilisant la procédure d'APPEL DE RO;
- c) pour la primitive de réponse de G-MODIFICATION ne contenant pas d'indicateur lié, la CMIPM doit:
  - construire une APDU confirmant l'opération de modification de gestion;
  - si les paramètres présents dans la primitive de réponse de G-MODIFICATION indiquent que l'opération a été exécutée correctement, envoyer l'APDU en utilisant la procédure de RÉSULTAT DE RO. Si les paramètres présents dans la primitive de réponse de G-MODIFICATION 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'APDU en utilisant la procédure d'ERREUR DE RO.

### 6.5.4 Réception de la réponse

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

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

## 6.6 Procédure d'action

### 6.6.1 Appel

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

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

## Remplacée par une version plus récente

- a) dans le mode confirmé, construire une APDU demandant l'opération d'action de gestion de type confirmé (mActionConfirmed), ou sinon, construire une APDU demandant l'opération d'action de gestion (mAction);
- b) envoyer l'APDU en utilisant la procédure d'APPEL DE RO.

### 6.6.2 Réception

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

### 6.6.3 Réponse

Dans le mode confirmé, la CMIPM doit:

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

### 6.6.4 Réception de la réponse

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

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

## 6.7 Procédure de création

### 6.7.1 Appel

Les procédures de création sont lancées par la primitive de demande de G-CRÉATION.

Dès réception de la primitive de demande de G-CRÉATION, la CMIPM doit:

- a) construire une APDU demandant l'opération de création de gestion (mCreate);
- b) envoyer l'APDU en utilisant la procédure d'APPEL DE RO.

## Remplacée par une version plus récente

### 6.7.2 Réception

Dès réception d'une APDU demandant l'opération de création de gestion, la CMIPM doit, si l'APDU est bien formée, envoyer une primitive d'indication de G-CRÉATION à l'utilisateur du service CMISE, ou sinon, construire une APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure de REFUS DE RO PAR L'UTILISATEUR.

### 6.7.3 Réponse

La CMIPM doit accepter une primitive de réponse de G-CRÉATION et doit:

- a) construire une APDU confirmant l'opération de création de gestion;
- b) si les paramètres présents dans la primitive de réponse de G-CRÉATION indiquent que l'opération a été exécutée correctement, envoyer l'APDU en utilisant la procédure de RÉSULTAT DE RO, ou sinon, envoyer l'APDU en utilisant la procédure d'ERREUR DE RO.

### 6.7.4 Réception de la réponse

Dès réception d'une APDU répondant à une opération de création de gestion, la CMIPM doit, si l'APDU est bien formée, envoyer une primitive de confirmation de G-CRÉATION à l'utilisateur du service CMISE, terminant ainsi la procédure de G-CRÉATION, ou sinon, construire une APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure de REFUS DE RO PAR L'UTILISATEUR.

## 6.8 Procédure de suppression

### 6.8.1 Appel

Les procédures de suppression sont lancées par la primitive de demande de G-SUPPRESSION.

Dès réception de la primitive de demande de G-SUPPRESSION, la CMIPM doit:

- a) construire une APDU demandant l'opération de G-SUPPRESSION (mDelete);
- b) envoyer l'APDU en utilisant la procédure d'APPEL DE RO.

### 6.8.2 Réception

Dès réception d'une APDU demandant l'opération de suppression de gestion, la CMIPM doit, si l'APDU est bien formée, envoyer une primitive d'indication de G-SUPPRESSION à l'utilisateur du service CMISE, ou sinon, construire une APDU contenant la notification de l'erreur et l'envoyer en utilisant la procédure de REFUS DE RO PAR L'UTILISATEUR.

### 6.8.3 Réponse

La CMIPM doit:

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

# Remplacée par une version plus récente

## 6.8.4 Réception de la réponse

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

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

## 6.9 Terminaison normale d'association

Chacun des utilisateurs du service CMISE peut lancer une terminaison normale d'association en recourant au service de TERMINAISON D'ASSOCIATION de la Recommandation X.217 [7].

*Remarque* – Cette spécification diffère de l'utilisation ROSE de l'opération BIND (liante) dans laquelle seul le demandeur de l'association peut utiliser la procédure de TERMINAISON D'ASSOCIATION.

## 6.10 Terminaison brusque d'association

Chacun des utilisateurs du service CMISE peut lancer une terminaison brusque d'association en utilisant le service de RUPTURE D'ASSOCIATION de la Recommandation X.217 [7].

Le fournisseur du service CMISE peut lancer une terminaison brusque de l'association en utilisant le service de RUPTURE D'ASSOCIATION PAR LE FOURNISSEUR de la Recommandation X.217 [7].

## 7 Syntaxe abstraite

Le présent paragraphe spécifie la syntaxe abstraite des PDU du CMIP.

### 7.1 Conventions

La syntaxe abstraite est définie à l'aide de la notation décrite dans la Recommandation X.208 [3]. Les productions MACRO de l'ASN.1 utilisées dans la présente Recommandation ou auxquelles il est fait référence ici ne tiennent pas compte des aspects ambigus de la grammaire.

A chacun des paramètres des services CMISE qui doit être transféré par une PDU du CMIP, correspond un champ de PDU (un type nommé (Namedtype) ASN.1), de même nom que ce paramètre (voir la Recommandation X.710 [12]), sous réserve des différences requises par l'utilisation des 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 sont abrégés comme suit:

ack	Accusé de réception (acknowledgement)
arg	Argument (argument)
id	Identificateur (identifier)
info	Information (information)
sync	Synchronisation (synchronization).

# Remplacée par une version plus récente

## 7.2 Correspondance entre les primitives du CMISE et les opérations du CMIP

TABLEAU 4/X.711

Correspondance entre les primitives du CMISE et les opérations du CMIP

Primitive CMIS	Mode	Identificateur lié	Opération du CMIP
Demande/indication de G-OBTENTION-ANNULATION	Confirmé	Non applicable	Obtention d'annulation de gestion de type confirmé ( <u>m-Cancel-Get-Confirmed</u> )
Réponse/confirmation de G-OBTENTION-ANNULATION	Non applicable	Non applicable	Obtention d'annulation de gestion de type confirmé ( <u>m-Cancel-Get-Confirmed</u> )
Demande/indication de G-NOTIFICATION-ÉVÈNEMENT	Non confirmé	Non applicable	Notification d'événement de gestion ( <u>m-EventReport</u> )
Demande/indication de G-NOTIFICATION-ÉVÈNEMENT	Confirmé	Non applicable	Notification d'événement de gestion de type confirmé ( <u>m-EventReport-Confirmed</u> )
Réponse à une demande/confirmation de G-NOTIFICATION-ÉVÈNEMENT	Non applicable	Non applicable	Notification d'événement de gestion de type confirmé ( <u>m-EventReport-Confirmed</u> )
Demande/indication de G-OBTENTION	Confirmé	Non applicable	Obtention de gestion ( <u>m-Get</u> )
Réponse à une demande/confirmation de G-OBTENTION	Non applicable	Absent	Obtention de gestion ( <u>m-Get</u> )
Réponse à une demande/confirmation de G-OBTENTION	Non applicable	Présent	Réponse liée de gestion ( <u>m-Linked-Reply</u> )
Demande/indication de G-MODIFICATION	Non confirmé	Non applicable	Modification de gestion ( <u>m-Set</u> )
Demande/indication de G-MODIFICATION	Confirmé	Non applicable	Modification de gestion de type confirmé ( <u>m-Set-Confirmed</u> )
Réponse à une demande/confirmation de G-MODIFICATION	Non applicable	Absent	Modification de gestion de type confirmé ( <u>m-Set-Confirmed</u> )
Réponse à une demande/confirmation de G-MODIFICATION	Non applicable	Présent	Réponse liée de gestion ( <u>m-Linked-Reply</u> )
Demande/indication de G-ACTION	Non confirmé	Non applicable	Action de gestion ( <u>m-Action</u> )
Demande/indication de G-ACTION	Confirmé	Non applicable	Action de gestion de type confirmé ( <u>m-Action-Confirmed</u> )
Réponse à une demande/confirmation de G-ACTION	Non applicable	Absent	Action de gestion de type confirmé ( <u>m-Action-Confirmed</u> )
Réponse à une demande/confirmation de G-ACTION	Non applicable	Présent	Réponse liée de gestion ( <u>m-Linked-Reply</u> )
Demande/indication de G-CRÉATION	Confirmé	Non applicable	Création de gestion ( <u>m-Create</u> )
Réponse à une demande/confirmation de G-CRÉATION	Non applicable	Non applicable	Création de gestion ( <u>m-Create</u> )
Demande/indication de G-SUPPRESSION	Confirmé	Non applicable	Suppression de gestion ( <u>m-Delete</u> )
Réponse à une demande/confirmation de G-SUPPRESSION	Non applicable	Absent	Suppression de gestion ( <u>m-Delete</u> )
Réponse à une demande/confirmation de G-SUPPRESSION	Non applicable	Présent	Réponse liée de gestion ( <u>m-Linked-Reply</u> )

Remarque – La mise en correspondance des macros OPERATION et ERROR avec le ROSE est définie dans la Recommandation X.219 [8].

## 7.3 Données de l'utilisateur ACSE

## Remplacée par une version plus récente

Le protocole ACSE (Recommandation X.227 [10]) est décrit en notation ASN.1. Les «informations de l'utilisateur» sont décrites en utilisant le type de données EXTERNAL.

### 7.3.1 Données de l'utilisateur d'ÉTABLISSEMENT D'ASSOCIATION

Le codage des informations de l'utilisateur CMIP à communiquer à l'ÉTABLISSEMENT D'ASSOCIATION dans le paramètre «informations de l'utilisateur» est défini comme suit:

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

**DEFINITIONS ::= BEGIN**

**FunctionalUnits ::= BIT STRING {**

<b>multipleObjectSelection</b>	<b>(0),</b>
<b>filter</b>	<b>(1),</b>
<b>multipleReply</b>	<b>(2),</b>
<b>extendedService</b>	<b>(3),</b>
<b>cancelGet</b>	<b>(4)</b>

**}**

-- 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 de l'utilisateur» de l'ÉTABLISSEMENT D'ASSOCIATION

**CMIPUserInfo ::= SEQUENCE {**

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

**}**

**ProtocolVersion ::= BIT STRING {**

<b>version1</b>	<b>(0),</b>
<b>version2</b>	<b>(1)</b>

**}**

*Remarque* – La présente Recommandation spécifie la version 2 du protocole pour la compatibilité technique avec l'ISO/CEI 9596-1: 1991.

**END**

Le codage des autres «informations de l'utilisateur» fournies par l'utilisateur du service CMISE n'est pas défini dans la présente Recommandation.

### 7.3.2 Données de l'utilisateur de RUPTURE D'ASSOCIATION

Le codage des informations de l'utilisateur CMIP à communiquer à la RUPTURE D'ASSOCIATION dans le paramètre «informations de l'utilisateur» est défini comme suit:

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

**DEFINITIONS ::= BEGIN**

-- Information acheminée dans le paramètre «informations de l'utilisateur» de RUPTURE D'ASSOCIATION

**CMIPAbortInfo ::= SEQUENCE {**

<b>abortSource</b>	<b>[0] IMPLICIT CMIPAbortSource,</b>
<b>userInfo</b>	<b>[1] EXTERNAL OPTIONAL</b>

**}**

## Remplacée par une version plus récente

```
CMIPAbortSource ::= ENUMERATED {  
    cmiseServiceUser          (0),  
    cmiseServiceProvider      (1)  
}
```

END

Le codage des autres «informations de l'utilisateur» fourni par l'utilisateur du service CMISE n'est pas défini dans la présente Recommandation.

### 7.4 Unités de données CMIP

Le protocole est décrit en terme d'unités de données du protocole commun de transfert d'informations de gestion, échangées entre les CMISE homologues. Les PDU sont spécifiées en utilisant la notation ASN.1 et les macro externes OPERATION et ERROR du protocole d'opérations distantes, définies dans la Recommandation X.219 [8].

-- *Protocole commun de transfert d'informations de gestion (CMIP)*

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

```
DEFINITIONS ::= BEGIN
```

-- *Définition des opérations distantes*

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

-- *Définition du service des opérations distantes*

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

-- *Définition du service d'annuaire*

```
DistinguishedName, RDNSequence 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'APDU ROSE correspondant*

-- *Opérations d'action (G-ACTION)*

```
m-Action OPERATION  
    ARGUMENT ActionArgument  
    ::= localValue 6
```

```
m-Action-Confirmed OPERATION  
    ARGUMENT      ActionArgument  
    RESULT        ActionResult
```

-- *Ce résultat est conditionnel; pour les conditions, voir le § 8.3.3.2.9 de la Recommandation X.710*

```
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
```

# Remplacée par une version plus récente

## RESULT

```
ERRORS { mistypedOperation, noSuchInvokeld, processingFailure }  
::= localValue 10
```

-- *Opération de création (G-CRÉATION)*

### m-Create OPERATION

```
ARGUMENT      CreateArgument  
RESULT        CreateResult
```

-- *Ce résultat est conditionnel; pour les conditions, voir le § 8.3.4.1.3 de la Recommandation X.710*

## ERRORS {

```
accessDenied, classInstanceConflict, duplicateManagedObjectInstance, invalidAttributeValue,  
invalidObjectInstance, missingAttributeValue, noSuchAttribute, noSuchObjectClass,  
noSuchObjectInstance, noSuchReferenceObject, processingFailure  
}  
::= localValue 8
```

-- *Opération de suppression (G-SUPPRESSION)*

### m-Delete OPERATION

```
Argument      DeleteArgument  
RESULT        DeleteResult
```

-- *Ce résultat est conditionnel; pour les conditions, voir le § 8.3.5.2.8 de la Recommandation X.710*

## ERRORS {

```
accessDenied, classInstanceConflict, complexityLimitation, invalidFilter,  
InvalidScope, noSuchObjectClass, noSuchObjectInstance, processingFailure, syncNotSupported  
}
```

## LINKED { m-Linked-Reply }

```
::= localValue 9
```

-- *Opération de notification d'événement (G-NOTIFICATION-ÉVÉNEMENT)*

### 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'obtention (G-OBTENTION)*

### m-Get OPERATION

```
ARGUMENT      GetArgument  
RESULT        GetResult
```

-- *Ce résultat est conditionnel; pour les conditions, voir le § 8.3.1.2.8 de la Recommandation X.710*

## ERRORS {

```
accessDenied, classInstanceConflict, complexityLimitation, getListError, invalidFilter, invalidScope,  
noSuchObjectClass, noSuchObjectInstance, operationCancelled, processingFailure,  
syncNotSupported  
}
```

## Remplacée par une version plus récente

```
LINKED { m-Linked-Reply }  
 ::= localValue 3
```

-- *Opération liée pour G-OBTENTION, G-MODIFICATION (de type confirmé), G-ACTION (de type confirmé) et G-SUPPRESSION*

```
m-Linked-Reply OPERATION  
 ARGUMENT      LinkedReplyArgument  
 ::= localValue 2
```

-- *Opération de modification (G-MODIFICATION)*

```
m-Set OPERATION  
 ARGUMENT      SetArgument  
 ::= localValue 4
```

```
m-Set-Confirmed OPERATION  
 ARGUMENT      SetArgument  
 RESULT        SetResult
```

-- *Ce résultat est conditionnel; pour les conditions, voir le § 8.3.2.2.9 de la Recommandation X.710*

```
ERRORS {  
   accessDenied, classInstanceConflict, complexityLimitation, invalidFilter, invalidScope,  
   noSuchObjectClass, noSuchObjectInstance, processingFailure, setListError, syncNotSupported  
 }
```

```
LINKED { m-Linked-Reply }  
 ::= localValue 5
```

-- *Définitions d'erreur du CMIS*

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

# Remplacée par une version plus récente

**invalidObjectInstance ERROR**  
PARAMETER      ObjectInstance  
::= localValue 17

**invalidScope ERROR**  
PARAMETER      Scope  
::= localValue 16

**missingAttributeValue ERROR**  
PARAMETER      SET OF Attributeld  
::= localValue 18

**mistypedOperation ERROR**  
::= localValue 21

**noSuchAction ERROR**  
PARAMETER      NoSuchAction  
::= localValue 9

**noSuchArgument ERROR**  
PARAMETER      NoSuchArgument  
::= localValue 14

**noSuchAttribute ERROR**  
PARAMETER      Attributeld  
::= localValue 5

**noSuchEventType ERROR**  
PARAMETER      NoSuchEventType  
::= localValue 13

**noSuchInvokeld ERROR**  
PARAMETER      InvokeldType  
::= 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

# Remplacée par une version plus récente

-- Définitions des types supports

```
AccessControl ::= EXTERNAL

ActionArgument ::= SEQUENCE {
    COMPONENTS OF
    accessControl
    synchronization
    scope
    filter
    actionInfo
}

ActionError ::= SEQUENCE {
    managedObjectClass
    managedObjectInstance
    currentTime
    actionErrorInfo
}

ActionErrorInfo ::= SEQUENCE {
    errorStatus
    errorInfo
}

errorStatus ::= ENUMERATED {
    accessDenied
    noSuchAction
    noSuchArgument
    invalidArgumentValue
},

errorInfo ::= CHOICE {
    actionType
    actionArgument
    argumentValue
}

ActionInfo ::= SEQUENCE {
    actionType
    actionInfoArg
}

ActionReply ::= SEQUENCE {
    actionType
    actionReplyInfo
}

ActionResult ::= SEQUENCE {
    managedObjectClass
    managedObjectInstance
    currentTime
    actionReply
}

ActionType ::= CHOICE {
    globalForm
    localForm
}
```

-- La présente Recommandation 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

```
Attribute ::= SEQUENCE {
    attributeld
    attributeValue
}
```

## Remplacée par une version plus récente

<b>AttributeError</b>	<b>::= SEQUENCE {</b>
<b>errorStatus</b>	<b>ENUMERATED {</b>
<b>accessDenied</b>	(2),
<b>noSuchAttribute</b>	(5),
<b>invalidAttributeValue</b>	(6),
<b>invalidOperation</b>	(24),
<b>invalidOperator</b>	(25)
<b>},</b>	
<b>modifyOperator</b>	<b>[2] IMPLICIT ModifyOperator OPTIONAL,</b>
	<b>-- présent pour invalidOperator et pour invalidOperation</b>
<b>attributeld</b>	<b>Attributeld,</b>
<b>attributeValue</b>	<b>ANY DEFINED BY attributeld OPTIONAL</b>
	<b>-- absent for setToDefault</b>
<b>}</b>	
<b>Attributeld</b>	<b>::= CHOICE {</b>
<b>globalForm</b>	<b>[0] IMPLICIT OBJECT IDENTIFIER,</b>
<b>localForm</b>	<b>[1] IMPLICIT INTEGER</b>
<b>}</b>	

-- La présente Recommandation 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

<b>AttributeldError</b>	<b>::= SEQUENCE {</b>
<b>errorStatus</b>	<b>ENUMERATED {</b>
<b>accessDenied</b>	(2),
<b>noSuchAttribute</b>	(5) <b>},</b>
<b>attributeld</b>	<b>Attributeld</b>
<b>}</b>	
<b>BaseManagedObjectld</b>	<b>::= SEQUENCE {</b>
<b>baseManagedObjectClass</b>	<b>ObjectClass,</b>
<b>baseManagedObjectInstance</b>	<b>ObjectInstance</b>
<b>}</b>	
<b>CMISFilter</b>	<b>::= CHOICE {</b>
<b>item</b>	<b>[8] FilterItem,</b>
<b>and</b>	<b>[9] IMPLICIT SET OF CMISFilter,</b>
<b>or</b>	<b>[10] IMPLICIT SET OF CMISFilter,</b>
<b>not</b>	<b>[11] CMISFilter</b>
<b>}</b>	
<b>CMISync</b>	<b>::= ENUMERATED {</b>
<b>bestEffort</b>	(0),
<b>atomic</b>	(1)
<b>}</b>	
<b>ComplexityLimitation</b>	<b>::= SET {</b>
<b>scope</b>	<b>[0] Scope OPTIONAL,</b>
<b>filter</b>	<b>[1] CMISFilter OPTIONAL,</b>
<b>sync</b>	<b>[2] CMISync OPTIONAL</b>
<b>}</b>	
<b>CreateArgument</b>	<b>::= SEQUENCE {</b>
<b>managedObjectClass</b>	<b>ObjectClass,</b>

## Remplacée par une version plus récente

<b>CHOICE {</b>	
<b>managedObjectInstance</b>	<b>ObjectInstance,</b>
<b>superiorObjectInstance</b>	<b>[8] ObjectInstance } OPTIONAL,</b>
<b>accessControl</b>	<b>[5] AccessControl OPTIONAL,</b>
<b>referenceObjectInstance</b>	<b>[6] ObjectInstance OPTIONAL,</b>
<b>attributeList</b>	<b>[7] IMPLICIT SET OF Attribute OPTIONAL</b>
<b>}</b>	
<b>CreateResult</b>	<b>::= SEQUENCE {</b>
<b>managedObjectClass</b>	<b>ObjectClass OPTIONAL,</b>
<b>managedObjectInstance</b>	<b>ObjectInstance OPTIONAL,</b>
	<b>-- doit être retourné si omis de Create Argument</b>
<b>currentTime</b>	<b>[5] IMPLICIT GeneralizedTime OPTIONAL,</b>
<b>attributeList</b>	<b>[6] IMPLICIT SET OF Attribute OPTIONAL</b>
<b>}</b>	
<b>DeleteArgument</b>	<b>::= SEQUENCE {</b>
<b>COMPONENTS OF</b>	<b>BaseManagedObjectId,</b>
<b>accessControl</b>	<b>[5] AccessControl OPTIONAL,</b>
<b>synchronization</b>	<b>[6] IMPLICIT CMISSync DEFAULT bestEffort,</b>
<b>scope</b>	<b>[7] Scope DEFAULT baseObject,</b>
<b>filter</b>	<b>CMISFilter DEFAULT and {}</b>
<b>}</b>	
<b>DeleteError</b>	<b>::= SEQUENCE {</b>
<b>managedObjectClass</b>	<b>ObjectClass OPTIONAL,</b>
<b>managedObjectInstance</b>	<b>ObjectInstance OPTIONAL,</b>
<b>currentTime</b>	<b>[5] IMPLICIT GeneralizedTime OPTIONAL,</b>
<b>deleteErrorInfo</b>	<b>[6] ENUMERATED { accessDenied (2)</b>
<b>}}</b>	
<b>DeleteResult</b>	<b>::= SEQUENCE {</b>
<b>managedObjectClass</b>	<b>ObjectClass OPTIONAL,</b>
<b>managedObjectInstance</b>	<b>ObjectInstance OPTIONAL,</b>
<b>currentTime</b>	<b>[5] IMPLICIT GeneralizedTime OPTIONAL</b>
<b>}</b>	
<b>EventReply</b>	<b>::= SEQUENCE {</b>
<b>eventType</b>	<b>EventTypeId,</b>
<b>eventReplyInfo</b>	<b>[8] ANY DEFINED BY eventType OPTIONAL</b>
<b>}</b>	
<b>EventReportArgument</b>	<b>::= SEQUENCE {</b>
<b>managedObjectClass</b>	<b>ObjectClass,</b>
<b>managedObjectInstance</b>	<b>ObjectInstance,</b>
<b>eventTime</b>	<b>[5] IMPLICIT GeneralizedTime OPTIONAL,</b>
<b>eventType</b>	<b>EventTypeId,</b>
<b>eventInfo</b>	<b>[8] ANY DEFINED BY eventType OPTIONAL</b>
<b>}</b>	
<b>EventReportResult</b>	<b>::= SEQUENCE {</b>
<b>managedObjectClass</b>	<b>ObjectClass OPTIONAL,</b>
<b>managedObjectInstance</b>	<b>ObjectInstance OPTIONAL,</b>
<b>currentTime</b>	<b>[5] IMPLICIT GeneralizedTime OPTIONAL,</b>
<b>eventReply</b>	<b>EventReply OPTIONAL</b>
<b>}</b>	
<b>EventTypeId</b>	<b>::= CHOICE {</b>
<b>globalForm</b>	<b>[6] IMPLICIT OBJECT IDENTIFIER,</b>
<b>localForm</b>	<b>[7] IMPLICIT INTEGER</b>
<b>}</b>	

## Remplacée par une version plus récente

-- La présente Recommandation 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

<b>FilterItem</b>	<b>::= CHOICE {</b>
equality	[0] IMPLICIT Attribute,
substrings	[1] IMPLICIT SEQUENCE OF CHOICE {
initialString	[0] IMPLICIT SEQUENCE {
	attributeld            Attributeld,
	string                ANY DEFINED BY attributeld },
anyString	[1] IMPLICIT SEQUENCE {
	attributeld            Attributeld,
	string                ANY DEFINED BY attributeld },
finalString	[2] IMPLICIT SEQUENCE {
	attributeld            Attributeld,
	string                ANY DEFINED BY attributeld } },
greaterOrEqual	[2] IMPLICIT Attribute,    -- valeur retenue ≥ valeur de l'attribut
lessOrEqual	[3] IMPLICIT Attribute,    -- valeur retenue ≥ valeur de l'attribut
present	[4] Attributeld,
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
}	
<b>GetArgument</b>	<b>::= SEQUENCE {</b>
<b>COMPONENTS OF</b>	<b>BaseManagedObjectId,</b>
accessControl	[5] AccessControl OPTIONAL,
synchronization	[6] IMPLICIT CMISSync DEFAULT bestEffort,
scope	[7] Scope DEFAULT baseObject,
filter	CMISFilter DEFAULT and {},
attributeldList	[12] IMPLICIT SET OF Attributeld OPTIONAL
}	
<b>GetInfoStatus</b>	<b>::= CHOICE {</b>
attributeldError	[0] IMPLICIT AttributeldError,
attribute	[1] IMPLICIT Attribute
}	
<b>GetListError</b>	<b>::= SEQUENCE {</b>
managedObjectClass	ObjectClass OPTIONAL,
managedObjectInstance	ObjectInstance OPTIONAL,
currentTime	[5] IMPLICIT GeneralizedTime OPTIONAL,
getInfoList	[6] IMPLICIT SET OF GetInfoStatus
}	
<b>GetResult</b>	<b>::= SEQUENCE {</b>
managedObjectClass	ObjectClass OPTIONAL,
managedObjectInstance	ObjectInstance OPTIONAL,
currentTime	[5] IMPLICIT GeneralizedTime OPTIONAL,
attributeList	[6] IMPLICIT SET OF Attribute OPTIONAL
}	

## Remplacée par une version plus récente

<b>InvalidArgumentValue</b> actionValue eventValue  }	::= CHOICE { [0] IMPLICIT ActionInfo, [1] IMPLICIT SEQUENCE { eventType EventTyped, eventInfo [8] ANY DEFINED BY eventType OPTIONAL } }
<b>LinkedReplyArgument</b> getResult getListError setResult setListError actionResult processingFailure deleteResult actionError deleteError }	::= CHOICE { [0] IMPLICIT GetResult, [1] IMPLICIT GetListError, [2] IMPLICIT SetResult, [3] IMPLICIT SetListError, [4] IMPLICIT ActionResult, [5] IMPLICIT ProcessingFailure, [6] IMPLICIT DeleteResult, [7] IMPLICIT ActionError, [8] IMPLICIT DeleteError }
<b>ModifyOperator</b> replace addValues removeValues setDefault }	::= INTEGER { (0), (1), (2), (3) }
<b>NoSuchAction</b> managedObjectClass actionType }	::= SEQUENCE { ObjectClass, ActionTypeld }
<b>NoSuchArgument</b> actionId  eventId  }	::= CHOICE { [0] IMPLICIT SEQUENCE { managedObjectClass ObjectClass OPTIONAL, actionType ActionTypeld }, [1] IMPLICIT SEQUENCE { managedObjectClass ObjectClass OPTIONAL, eventType EventTyped } }
<b>NoSuchEventType</b> managedObjectClass eventType }	::= SEQUENCE { ObjectClass, EventTypeld }
<b>ObjectClass</b> globalForm localForm }	::= CHOICE { [0] IMPLICIT OBJECT IDENTIFIER, [1] IMPLICIT INTEGER }

-- La présente Recommandation 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

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

## Remplacée par une version plus récente

-- *local Distinguished Name 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 { INTEGER {
    baseObject              (0),
    firstLevelOnly         (1),
    wholeSubtree            (2) },
    [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 de 0 a la même sémantique que baseObjet*  
-- *avec individualLevels, une valeur de 1 a la même sémantique que firstLevelOnly*

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

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

ListError ::= 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 CMIP
```

# Remplacée par une version plus récente

## 7.5 Définition de la syntaxe abstraite du CMIP

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

**{joint-iso-ccitt ms(9) cmip(1) cmip-pci(1) abstractSyntax(4)}**

comme nom de syntaxe abstraite de 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 Recommandation X.229 [11], le composant argument rempli selon la définition de CMIP-1, ou une valeur de l'un des types ASN.1:

- **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 lesquels X est du type IDENTIFICATEUR D'OBJET.

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 Recommandation X.209 [4]) peuvent être utilisées comme nom de syntaxe de transfert avec cette syntaxe abstraite.

### 7.5.1 Règles d'extensibilité

7.5.1.1 La CMIPM accepteur doit, lorsqu'il traite une information de type CMIP-A-ASSOCIATE:

- ne tenir compte d'aucune valeur d'étiqueté non définie dans la syntaxe abstraite de cette Recommandation; et
- ne tenir compte d'aucune affectation de non 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 du système;
- 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 déclaré mettre en œuvre les procédures spécifiées dans la présente norme doit se conformer aux spécifications des § 8.1 et 8.2.

### 8.1 Conditions de conformité statique

Le système doit:

- a) prendre en charge l'unité fonctionnelle «noyau» définie dans la Recommandation X.710 [12] et les facilités impliquées par cette unité fonctionnelle;
- b) prendre en charge la syntaxe de transfert dérivée des règles de codage spécifiées dans la Recommandation X.209 [4] et nommée:

# Remplacée par une version plus récente

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

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

“CMIP-PCI”

pour les unités fonctionnelles prises en charge;

- c) prendre en charge le protocole ACSE défini dans la Recommandation X.227 [10] pour établir et terminer une association;
- d) prendre en charge les règles spécifiées dans l'annexe A dans tout contexte d'application comptant le CMISE parmi l'un des ASE;
- e) prendre en charge la classe d'associations 3 du protocole ROSE défini dans la Recommandation X.229 [11];
- f) prendre en charge l'unité fonctionnelle de réponses multiples si l'unité fonctionnelle de sélection d'objets multiples est prise en charge.

## 8.2 Conditions de conformité dynamique

Le système doit:

- a) suivre les procédures relevant de chaque unité fonctionnelle que le système est déclaré mettre en œuvre;
- b) s'ils sont utilisés, vérifier les paramètres de sécurité facultatifs définis dans les PDU du CMIP;
- c) si l'unité fonctionnelle d'extension de services est prise en charge, prendre en charge le protocole de présentation défini dans la Recommandation X.226 [9], comme requis par le contexte d'application;
- d) si le masquage est assuré, prendre en charge l'unité fonctionnelle de réponses multiples.

## ANNEXE A

(ne fait pas partie intégrante de cette Recommandation)

### Règles d'association pour le CMISE

#### A.1 ACSE, caractéristiques de session et de présentation

A.1.1 Le CMISE exige l'unité fonctionnelle de présentation «noyau» qui est définie dans la Recommandation X.216 [6].

A.1.2 Le CMISE exige les unités fonctionnelles de session «noyau» et «transmission duplex» qui sont définies dans la Recommandation X.215 [5].

A.1.3 Le CMISE exige le mode normal des services ACSE et de présentation qui sont définis dans les Recommandation X.227 [10] et X.216 [6].

#### 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 de l'utilisateur» ÉTABLISSEMENT D'ASSOCIATION que définit la Recommandation X.710 [12]. Les informations de l'utilisateur CMIP doivent être mises à la disposition de la CMIPM qui doit:

## Remplacée par une version plus récente

- a) construire CMIPUserInfo à partir des informations fournies;
- b) placer le paramètre de la version de protocole dans CMIPUserInfo en indiquant le bit correspondant à chaque version prise en charge;
- c) insérer CMIPUserInfo en tant qu'informations EXTERNES (EXTERNAL) distinctes dans le paramètre «informations de l'utilisateur» de la primitive de demande d'ÉTABLISSEMENT D'ASSOCIATION;
- d) attendre que les informations de l'utilisateur qui sont spécifiques au CMIS soient renvoyées dans la primitive de confirmation d'ÉTABLISSEMENT D'ASSOCIATION.

### A.2.2 Indication

Dès réception d'une primitive d'indication d'ÉTABLISSEMENT D'ASSOCIATION, le paramètre CMIPUserInfo doit être mis à la disposition de la 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 d'ÉTABLISSEMENT D'ASSOCIATION étant mis sur «refusé par le répondeur (permanent)». L'association n'est pas établie et cette occurrence de la 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 de l'utilisateur. La CMIPM doit attendre la réponse de l'utilisateur du service CMISE.

### A.2.3 Réponse

La primitive de réponse d'ÉTABLISSEMENT D'ASSOCIATION 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 CMIPM qui doit:

- a) construire CMIPUserInfo qui est nécessaire pour la réponse. CMIPUserInfo doit comprendre le paramètre de la version indiquant toutes les versions du CMIP qui sont prises en charge;
- b) insérer CMIPUserInfo en tant qu'informations EXTERNES distinctes dans le paramètre «informations de l'utilisateur» de la primitive de réponse ÉTABLISSEMENT D'ASSOCIATION;
- 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 CMIPM. La CMIPM est alors prête à accepter les primitives d'indication de CMISE;
- d) si la réponse d'association indique «refusé», cette occurrence de la CMIPM doit cesser d'exister.

### A.2.4 Confirmation

Dès réception de la primitive de confirmation d'ÉTABLISSEMENT D'ASSOCIATION, le paramètre CMIPUserInfo doit être mis à la disposition de la CMIP 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 de l'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 occurrence de la CMIPM doit cesser d'exister.

### A.3 Règles de terminaison d'association

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

# Remplacée par une version plus récente

## A.3.1 *Demande*

Sur réception d'une demande de terminaison d'association, les paramètres nécessaires de TERMINAISON D'ASSOCIATION doivent être mis à la disposition de la CMIPM qui doit cesser d'accepter les demandes de service et attendre la confirmation de la terminaison de l'association.

## A.3.2 *Indication*

Sur réception d'une primitive d'indication de TERMINAISON D'ASSOCIATION, les paramètres nécessaires d'indication de TERMINAISON D'ASSOCIATION doivent être mis à la disposition de la CMIPM acceptante qui doit attendre la réponse de terminaison d'association.

## 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 de TERMINAISON D'ASSOCIATION doivent être mis à la disposition de la CMIPM acceptante. Par la suite, cette occurrence de la CMIPM doit cesser d'exister.

## A.3.4 *Confirmation*

Dès réception d'une primitive de confirmation de TERMINAISON D'ASSOCIATION, les paramètres nécessaires de confirmation de TERMINAISON D'ASSOCIATION doivent être mis à la disposition de la CMIPM requérante. Par la suite, cette occurrence de la 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 de RUPTURE D'ASSOCIATION, le paramètre CMIPAbortInfo ne peut être inclus dans les primitives du service de RUPTURE D'ASSOCIATION.

### A.4.1 *Demande de RUPTURE D'ASSOCIATION*

Dès réception d'une demande de rupture d'association, les paramètres nécessaires de demande de RUPTURE D'ASSOCIATION comprenant les informations de l'utilisateur de RUPTURE D'ASSOCIATION définies dans la Recommandation X.710 [12] doivent être mis à la disposition de la 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 de l'utilisateur» de la primitive de demande de RUPTURE D'ASSOCIATION;
- d) par la suite, cette occurrence de la CMIPM doit cesser d'exister.

### A.4.2 *Indication de RUPTURE D'ASSOCIATION*

Dès réception d'une primitive d'indication de RUPTURE D'ASSOCIATION, les paramètres nécessaires d'indication de RUPTURE D'ASSOCIATION comprenant CMIPAbortInfo doivent être mis à la disposition de la CMIPM. Par la suite, cette occurrence de la CMIPM doit cesser d'exister.

### A.4.3 *Indication de RUPTURE D'ASSOCIATION PAR LE FOURNISSEUR*

Dès réception d'une primitive d'indication de RUPTURE D'ASSOCIATION PAR LE FOURNISSEUR, les paramètres nécessaires d'indication de RUPTURE D'ASSOCIATION PAR LE FOURNISSEUR doivent être mis à la disposition de la CMIPM. Par la suite, cette occurrence de la CMIPM doit cesser d'exister.

# Remplacée par une version plus récente

## A.4.4 Erreur de protocole CMIP

Dès qu'elle détecte une erreur de protocole, la 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 de l'utilisateur» de la primitive de demande de RUPTURE D'ASSOCIATION;
- d) par la suite, cette occurrence de la CMIPM doit cesser d'exister.

## ANNEXE B

(ne fait pas partie intégrante de cette Recommandation)

### Extension de la syntaxe ASN.1

La présente annexe indique de quelle façon les macros OPERATION (opération) et ERROR (erreur) de la Recommandation X.219 [8] sont développées en types et sous-types de données ASN.1.

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

-- Protocole commun de transfert d'informations de gestion (CMIP)

**CMIP-1 {joint-iso-ccitt 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 remoteOperations(4) notation(0)}**

-- Définitions du service d'annuaire

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

-- Opérations du 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
}
```

## Remplacée par une version plus récente

```
ROERapdu ::= SEQUENCE {
    invokeID
    error-value
    parameter
}
InvokeIDType,
ERROR,
ANY DEFINED BY error-value OPTIONAL
```

```
RORJapdu ::= SEQUENCE {
    invokeID CHOICE {
    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 ROSE
    mistypedAPDU (1),
    badlyStructuredAPDU (2)
}
```

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

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

```
ReturnErrorProblem ::= INTEGER {
    unrecognisedInvocation (0), -- détecté par l'utilisateur 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 CMIP

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

## Remplacée par une version plus récente

- Cette partie de la spécification ASN.1 fournit une définition des sous-types apduROIV et apduRORS utilisés par le CMIP. Les sous-types de l'apduROIV 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'apduRORS 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 CMIP

<b>m-Action OPERATION</b>	<b>::= localValue 6</b>
<b>ROIV-m-Action</b> { invokeID linked-ID operation-value argument	<b>::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-Action), (INCLUDES ActionArgument) ) )</b>
<b>m-Action-Confirmed OPERATION</b>	<b>::= localValue 7</b>
<b>ROIV-m-Action-Confirmed</b> { invokeID linked-ID operation-value argument	<b>::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-Action-Confirmed), (INCLUDES ActionArgument) ) )</b>
<b>RORS-m-Action-Confirmed</b> { ... , invokeID <i>-- séquence de résultat -- (WITH COMPONENTS</i> { operation-value result <i>-- requis uniquement s'il existe une seule réponse à l'apduROIV d'action m ROIV de type confirmé (ROIV-m-Action-Confirmed ROIVapdu) et les données doivent être renvoyées dans l'apduRORS</i> })	<b>::= RORSapdu (WITH COMPONENTS PRESENT, (m-Action-Confirmed), (INCLUDES ActionResult) ) )</b>
<b>m-Cancel-Get OPERATION</b>	<b>::= localValue 10</b>
<b>ROIV-m-Cancel-Get</b> { invokeID linked-ID operation-value argument })	<b>::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-Cancel-Get), (INCLUDES InvokeIDType)</b>
<b>RORS-m-Cancel-Get</b> { invokeID <i>-- il n'y a pas de séquence de résultat pour RORS-m-Cancel-Get</i> })	<b>::= RORSapdu (WITH COMPONENTS PRESENT,</b>
<b>m-Create OPERATION</b>	<b>::= localValue 8</b>
<b>ROIV-m-Create</b> { invokeID linked-ID operation-value argument	<b>::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-Create), (INCLUDES CreateArgument) ) )</b>
<b>RORS-m-Create</b> { ... , invoke-ID <i>-- séquence de résultat -- (WITH COMPONENTS</i> { operation-value result })	<b>::= RORSapdu (WITH COMPONENTS PRESENT, (m-Create), (INCLUDES CreateResult) ) )</b>

## Remplacée par une version plus récente

<b>m-Delete OPERATION</b>	<b>::= localValue 9</b>
<b>ROIV-m-Delete</b> { invokeID linked-ID operation-value argument	<b>::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-Delete), (INCLUDES DeleteArgument) } )</b>
<b>RORS-m-Delete</b> { ... , invokeID <i>-- séquence de résultat -- (WITH COMPONENTS</i> { operation-value result <i>-- requis uniquement s'il existe une seule réponse à l'apduROIV de suppression m ROIV (ROIV-m-DeleteROIVapdu) et les données doivent être renvoyées dans l'apduRORS</i> } )	<b>::= RORSapdu (WITH COMPONENTS PRESENT, (m-Delete), (INCLUDES DeleteResult) } )</b>
<b>m-EventReport OPERATION</b>	<b>::= localValue 0</b>
<b>ROIV-m-EventReport</b> { invokeID linked-ID operation-value argument	<b>::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-EventReport), (INCLUDES EventReportArgument) } )</b>
<b>m-EventReport-Confirmed OPERATION</b>	<b>::= localValue 1</b>
<b>ROIV-m-EventReport-Confirmed</b> { invokeID linked-ID operation-value argument	<b>::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-EventReport-Confirmed), (INCLUDES EventReportArgument) } )</b>
<b>RORS-m-EventReport-Confirmed::= RORSapdu (WITH COMPONENTS</b> { ... , invokeID <i>-- séquence de résultat -- (WITH COMPONENTS</i> { operation-value result <i>-- requis uniquement si les données doivent être renvoyées dans l'apdurors</i> } )	<b>PRESENT, (m-EventReport-Confirmed), (INCLUDES EventReportResult) } )</b>
<b>m-Get OPERATION</b>	<b>::= localValue 3</b>
<b>ROIV-m-Get</b> { invokeID linked-ID operation-value argument	<b>::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-Get), (INCLUDES GetArgument) } )</b>
<b>RORS-m-Get</b> { ... , invokeID <i>-- séquence de résultat -- (WITH COMPONENTS</i> { operation-value result <i>-- requis uniquement s'il existe une seule réponse à l'apduROIV</i> } )	<b>::= RORSapdu (WITH COMPONENTS PRESENT, (m-Get), (INCLUDES GetResult) } )</b>

## Remplacée par une version plus récente

<b>m-Linked-Reply OPERATION</b>	<b>::= localValue 2</b>
<b>ROIV-m-Linked-Reply</b> { invokeID linked-ID operation-value argument       }	<b>::= ROIVapdu (WITH COMPONENTS          PRESENT,          PRESENT,          (m-Linked-Reply),          (INCLUDES LinkedReplyArgument) } )</b>

-- 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 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

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

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

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

## Remplacée par une version plus récente

```

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

```

```

m-Set-Confirmed OPERATION ::= localValue 5

```

```

ROIV-m-Set-Confirmed ::= ROIVapdu (WITH COMPONENTS
{ invokeID
  linked-ID
  operation-value
  argument
  PRESENT,
  ABSENT,
  (m-Set-Confirmed),
  (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'apduROIV de modification m ROIV en mode confirme
  (ROIV-mSet-Confirmed ROIVapdu)
  } )

```

-- Cette partie de la spécification ASN.1 fournit une définition des sous-types d'apduROER utilisés par le CMIP. Les sous-types de l'apduROER 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
  error-value
  PRESENT,
  (accessDenied) } )

```

-- Cette apduROER peut seulement être renvoyée en réponse aux apdu obtention-m-ROIV, modification-m-ROIV de type confirmé, action-m-ROIV de type confirmé, création-m-ROIV et suppression-m-ROIV

```

classInstanceConflict ERROR ::= localValue 19

```

```

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

```

## Remplacée par une version plus récente

-- Cette apduROER peut seulement être renvoyée en réponse aux apdu obtention-m-ROIV, modification-m-ROIV de type confirmé, action-m-ROIV de type confirmé, création-m-ROIV et ROIV suppression-m-ROIV

```
complexityLimitation ERROR ::= localValue 20

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

-- Cette apduROER peut seulement être renvoyée en réponse aux apdu obtention-m-ROIV, modification-m-ROIV de type confirmé, action-m-ROIV de type confirmé, et ROIV suppression-m-ROIV

```
duplicateManagedObjectInstance ERROR ::= localValue 11

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

-- Cette apduROER peut seulement être renvoyée en réponse à l'apduROIV de création-m-ROIV (ROIV-m-Create ROIVapdu)

```
getListError ERROR ::= localValue 7

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

-- Cette apduROER peut seulement être renvoyée en réponse à l'apduROIV d'obtention-m-ROIV (ROIV-m-Get ROIVapdu)

```
invalidArgumentValue ERROR ::= localValue 15

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

-- Cette apduROER peut seulement être renvoyée en réponse aux apdu notification d'événement-m-ROIV de type confirmé (ROIV-m-EventReport-Confirmed) et ROIV-d'action-m-ROIV de type confirmé (ROIV-m-Action-Confirmed ROIVapdus)

```
invalidAttributeValue ERROR ::= localValue 6

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

-- Cette apduROER peut seulement être renvoyée en réponse à l'apduROIV de création-m-ROIV (ROIV-m-Create ROIVapdu)

```
invalidFilter ERROR ::= localValue 4

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

## Remplacée par une version plus récente

- Cette apduROER peut seulement être renvoyée en réponse aux apdu obtention-m-ROIV, modification-m-ROIV de type confirmé, action-m-ROIV de type confirmé et ROIV-suppression-m-ROIV
- |   |  |
|---|--|
| <b>invalidObjectInstance ERROR</b>  | <b>::= localValue 17</b>   |
| <b>ROER-invalidObjectInstance</b><br>{ invokeID<br>error-value<br>parameter | <b>::= ROERapdu (WITH COMPONENTS<br/>PRESENT,<br/>(invalidObjectInstance),<br/>(INCLUDES ObjectInstance) )</b> |
- Cette apduROER peut seulement être renvoyée en réponse à l'apduROIV de création-m-ROIV
- |  |  |
|--|--|
| <b>invalidScope ERROR</b>  | <b>::= localValue 16</b>   |
| <b>ROER-invalidScope</b><br>{ invokeID<br>error-value<br>parameter | <b>::= ROERapdu (WITH COMPONENTS<br/>PRESENT,<br/>(invalidScope),<br/>(INCLUDES Scope) )</b> |
- Cette apduROER peut seulement être renvoyée en réponse aux apdu obtention-m-ROIV, modification-m-ROIV de type confirmé, action-m-ROIV de type confirmé et ROIV-suppression-m-ROIV
- |   |  |
|---|--|
| <b>missingAttributeValue ERROR</b>  | <b>::= localValue 18</b>   |
| <b>ROER-missingAttributeValue</b><br>{ invokeID<br>error-value<br>parameter | <b>::= ROERapdu (WITH COMPONENTS<br/>PRESENT,<br/>(missingAttributeValue),<br/>(INCLUDES SET OF Attributeld) )</b> |
- Cette apduROER peut seulement être renvoyée en réponse à l'apduROIV de création-m-ROIV
- |  |   |
|--|---|
| <b>mistypedOperation ERROR</b>                             | <b>::= localValue 21</b>  |
| <b>ROER-mistypedOperation</b><br>{ invokeID<br>error-value | <b>::= ROERapdu (WITH COMPONENTS<br/>PRESENT,<br/>(mistypedOperation) )</b> |
- Cette apduROER peut seulement être renvoyée en réponse à l'apduROIV d'obtention d'annulation-m-ROIV
- |  |   |
|--|---|
| <b>noSuchAction ERROR</b>  | <b>::= localValue 9</b>   |
| <b>ROER-noSuchAction</b><br>{ invokeID<br>error-value<br>parameter | <b>::= ROERapdu (WITH COMPONENTS<br/>PRESENT,<br/>(noSuchAction),<br/>(INCLUDES NoSuchAction) )</b> |
- Cette apduROER peut seulement être renvoyée en réponse à l'apdu ROIV d'action-m-ROIV de type confirmé
- |  |   |
|--|---|
| <b>noSuchArgument ERROR</b>  | <b>::= localValue 14</b>  |
| <b>ROER-noSuchArgument</b><br>{ invokeID<br>error-value<br>parameter | <b>::= ROERapdu (WITH COMPONENTS<br/>PRESENT,<br/>(noSuchArgument),<br/>(INCLUDES NoSuchArgument) )</b> |
- Cette apduROER peut seulement être renvoyée en réponse aux apdu notification d'événement-m-ROIV de type confirmé et ROIV d'action-m-ROIV de type confirmé
- |   |   |
|---|---|
| <b>noSuchAttribute ERROR</b>  | <b>::= localValue 5</b>   |
| <b>ROER-noSuchAttribute</b><br>{ invokeID<br>error-value<br>parameter | <b>::= ROERapdu (WITH COMPONENTS<br/>PRESENT,<br/>(noSuchAttribute),<br/>(INCLUDES Attributeld) )</b> |

## Remplacée par une version plus récente

-- Cette apduROER peut seulement être renvoyée en réponse à l'apduROIV de création-m-ROIV

<b>noSuchEventType ERROR</b>	<b>::= localValue 13</b>
<b>ROER-noSuchEventType</b> { invokeID error-value parameter	<b>::= ROERapdu (WITH COMPONENTS PRESENT, (noSuchEventType), (INCLUDES NoSuchEventType) } )</b>

-- Cette apduROER peut seulement être renvoyée en réponse à l'apduROIV de notification d'événement-m-ROIV de type confirmé

<b>noSuchInvokeld ERROR</b>	<b>::= localValue 22</b>
<b>ROER-noSuchInvokeld</b> { invokeID error-value parameter	<b>::= ROERapdu (WITH COMPONENTS PRESENT, (noSuchInvokeld), (INCLUDES InvokeldType) } )</b>

-- Cette apduROER peut seulement être renvoyée en réponse à l'apduROIV d'obtention d'annulation-m-ROIV

<b>noSuchObjectClass ERROR</b>	<b>::= localValue 0</b>
<b>ROER-noSuchObjectClass</b> { invokeID error-value parameter	<b>::= ROERapdu (WITH COMPONENTS PRESENT, (noSuchObjectClass ), (INCLUDES ObjectClass) } )</b>

-- Cette apduROER peut seulement être renvoyée en réponse aux apdu notification d'événement-m-ROIV de type confirmé, obtention-m-ROIV, modification-m-ROIV de type confirmé, action-m-ROIV de type confirmé, création-m-ROIV et ROIV-suppression-m-ROIV

<b>noSuchObjectInstance ERROR</b>	<b>::= localValue 1</b>
<b>ROER-noSuchObjectInstance</b> { invokeID error-value parameter	<b>::= ROERapdu (WITH COMPONENTS PRESENT, (noSuchObjectInstance), (INCLUDES ObjectInstance) } )</b>

-- Cette apduROER peut seulement être renvoyée en réponse aux apdu notification d'événement-m-ROIV obtention-m-ROIV, modification-m-ROIV de type confirmé, action-m-ROIV de type confirmé, création-m-ROIV et ROIV-suppression-m-ROIV

<b>noSuchReferenceObject ERROR</b>	<b>::= localValue 12</b>
<b>ROER-noSuchReferenceObject</b> { invokeID error-value parameter	<b>::= ROERapdu (WITH COMPONENTS PRESENT, (noSuchReferenceObject), (INCLUDES ObjectInstance) } )</b>

-- Cette apduROER peut seulement être renvoyée en réponse à l'apduROIV de création-m-ROIV

<b>operationCancelled ERROR</b>	<b>::= localValue 23</b>
<b>ROER-operationCancelled</b> { invokeID error-value	<b>::= ROERapdu (WITH COMPONENTS PRESENT, (operationCancelled) } )</b>

-- Cette apduROER peut seulement être renvoyée en réponse à l'apduROIV d'obtention-m-ROIV

<b>processingFailure ERROR</b>	<b>::= localValue 10</b>
<b>ROER-processingFailure</b> { invokeID error-value parameter	<b>::= ROERapdu (WITH COMPONENTS PRESENT, (processingFailure), (INCLUDES ProcessingFailure) OPTIONAL } )</b>

## Remplacée par une version plus récente

-- Cette apduROER peut seulement être renvoyée en réponse aux apdu notification d'événement-m-ROIV de type confirmé, obtention-m-ROIV, modification-m-ROIV de type confirmé, action-m-ROIV de type confirmé, création-m-ROIV et ROIV-suppression-m-ROIV

```
setListError ERROR ::= localValue 8

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

-- Cette apduROER peut seulement être renvoyée en réponse à l'apduROIV de modification-m-ROIV de type confirmé

```
syncNotSupported ERROR ::= localValue 3

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

-- Cette apduROER peut seulement être renvoyée en réponse aux apdu obtention-m-ROIV, modification-m-ROIV de type confirmé, action-m-ROIV de type confirmé, et ROIV-suppression-m-ROIV

-- 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 CMIP

### ANNEXE C

(ne fait pas partie intégrante de cette Recommandation)

#### Exemples d'APDU ROSE du CMISE

La présente annexe fournit quelques exemples du développement complet des APDU ROSE acheminant l'information CMIP.

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

-- adpuROIV 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 CMISync OPTIONAL,
    scope [7] Scope DEFAULT baseObject,
    filter CMISFilter DEFAULT and {},
    actionInfo [12] IMPLICIT SEQUENCE (
    actionType ActionType,
    actionInfoArg [4] ANY DEFINED BY actionType OPTIONAL
  ) } }
```

## Remplacée par une version plus récente

-- apduRORS pour l'opération d'action CMISE de type confirmé

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

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

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

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

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