



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Amendement 1

X.880

(11/95)

**RÉSEAUX DE COMMUNICATION DE DONNÉES
ET COMMUNICATION ENTRE SYSTÈMES OUVERTS
APPLICATIONS OSI – OPÉRATIONS DISTANTES**

**TECHNOLOGIES DE L'INFORMATION –
OPÉRATIONS DISTANTES:
CONCEPTS, MODÈLE ET NOTATION**

AMENDEMENT 1: OPÉRATIONS INTÉGRÉES

**Amendement 1 à la
Recommandation UIT-T X.880**

(Antérieurement «Recommandation du CCITT»)

AVANT-PROPOS

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

L'approbation des Recommandations par les membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la Conférence mondiale de normalisation des télécommunications (CMNT), (Helsinki, 1993). De plus, la CMNT, qui se réunit tous les quatre ans, approuve les Recommandations qui lui sont soumises et établit le programme d'études pour la période suivante.

Dans certains secteurs de la technologie de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI. Le texte de la Recommandation X.880, amendement 1, de l'UIT-T a été approuvé le 21 novembre 1995. Son texte est publié, sous forme identique, comme Norme internationale ISO/CEI 13712-1.

NOTE

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

© UIT 1996

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

RECOMMANDATIONS UIT-T DE LA SÉRIE X

**RÉSEAUX DE COMMUNICATION DE DONNÉES ET COMMUNICATION
ENTRE SYSTÈMES OUVERTS**

(Février 1994)

ORGANISATION DES RECOMMANDATIONS DE LA SÉRIE X

Domaine	Recommandations
RÉSEAUX PUBLICS POUR DONNÉES	
Services et services complémentaires	X.1-X.19
Interfaces	X.20-X.49
Transmission, signalisation et commutation	X.50-X.89
Aspects réseau	X.90-X.149
Maintenance	X.150-X.179
Dispositions administratives	X.180-X.199
INTERCONNEXION DES SYSTÈMES OUVERTS	
Modèle et notation	X.200-X.209
Définition des services	X.210-X.219
Spécifications des protocoles en mode connexion	X.220-X.229
Spécifications des protocoles en mode sans connexion	X.230-X.239
Formulaires PICS	X.240-X.259
Identification des protocoles	X.260-X.269
Protocoles de sécurité	X.270-X.279
Objets gérés de couche	X.280-X.289
Test de conformité	X.290-X.299
INTERFONCTIONNEMENT DES RÉSEAUX	
Considérations générales	X.300-X.349
Systèmes mobiles de transmission de données	X.350-X.369
Gestion	X.370-X.399
SYSTÈMES DE MESSAGERIE	X.400-X.499
ANNUAIRE	X.500-X.599
RÉSEAUTAGE OSI ET ASPECTS DES SYSTÈMES	
Réseautage	X.600-X.649
Dénomination, adressage et enregistrement	X.650-X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680-X.699
GESTION OSI	X.700-X.799
SÉCURITÉ	X.800-X.849
APPLICATIONS OSI	
Engagement, concomitance et rétablissement	X.850-X.859
Traitement des transactions	X.860-X.879
Opérations distantes	X.880-X.899
TRAITEMENT OUVERT RÉPARTI	X.900-X.999

TABLE DES MATIÈRES

	<i>Page</i>
1) Paragraphe 3.3.....	1
2) Paragraphe 8.2.1.....	1
3) Paragraphe 8.2.....	2
4) Paragraphe 10.1.....	2
5) Paragraphe 10.5.1.....	2
6) Paragraphe 10.5.2.....	2
7) Paragraphes 10.6 à 10.16	2
8) Paragraphes 10.6 à 10.11	2
9) Annexe A	4
10) Annexe D	6

Résumé

Le présent amendement à la Rec. X.880 | ISO/CEI 13712-1 définit trois opérations intégrées – sondage, acquittement et annulation – d'intérêt général pour les concepteurs d'applications basées sur l'élément de service d'opérations distantes (ROSE) (*remote operations service element*).

NORME INTERNATIONALE

RECOMMANDATION UIT-T

TECHNOLOGIES DE L'INFORMATION – OPÉRATIONS DISTANTES: CONCEPTS, MODÈLE ET NOTATION

AMENDEMENT 1 Opérations intégrées

1) Paragraphe 3.3

Ajouter la nouvelle définition suivante après 3.3.7:

«**3.3.8 idempotent**: caractéristique d'une opération qui peut être invoquée à plusieurs reprises sans modifier l'état de l'exécutant.»

Renommer en conséquence les définitions qui viennent à la suite du point 3.3.8.

2) Paragraphe 8.2.1

Ajouter le champ souligné suivant à la classe d'objets informationnels OPERATION:

```

OPERATION ::= CLASS
{
    &ArgumentType           OPTIONAL,
    &argumentTypeOptional   BOOLEAN OPTIONAL,
    &returnResult           BOOLEAN DEFAULT TRUE,
    &ResultType             OPTIONAL,
    &resultTypeOptional     BOOLEAN OPTIONAL,
    &Errors                 ERROR OPTIONAL,
    &Linked                 OPERATION OPTIONAL,
    &synchronous           BOOLEAN DEFAULT FALSE,
    &idempotent             BOOLEAN DEFAULT FALSE,
    &alwaysReturns         BOOLEAN DEFAULT TRUE,
    &InvokePriority         Priority OPTIONAL,
    &ResultPriority         Priority OPTIONAL,
    &operationCode         Code UNIQUE OPTIONAL
}
WITH SYNTAX
{
    [ARGUMENT           &ArgumentType [OPTIONAL   &argumentTypeOptional]]
    [RETURN RESULT     &returnResult]
    [RESULT            &ResultType [OPTIONAL   &resultTypeOptional]]
    [ERRORS           &Errors]
    [LINKED           &Linked]
    [SYNCHRONOUS     &synchronous]
    [IDEMPOTENT       &idempotent]
    [ALWAYS RESPONDS &alwaysReturns]
    [INVOKE PRIORITY &InvokePriority]
    [RESULT-PRIORITY &ResultPriority]
    [CODE             &operationCode]
}

```

3) **Paragraphe 8.2**

Ajouter le nouveau paragraphe qui suit:

«**8.2.14** Le champ `&idempotent` (*idempotent*) spécifie si l'opération est idempotente, il prend la valeur `TRUE` (*vrai*) si c'est le cas et la valeur `FALSE` (*faux*) dans le cas contraire.»

4) **Paragraphe 10.1**

Le point a) devient le suivant (les adjonctions sont soulignées):

«a) des opérations d'intérêt général (`emptyBind`, `emptyUnbind`, `no-op`, `probe`, `acknowledge`, `cancel`) avec leurs erreurs associées;»

5) **Paragraphe 10.5.1**

Ajouter le champ souligné comme suit à la définition de l'opération `no-op`:

```
no-op OPERATION ::=
{
  IDEMPOTENT      TRUE
  ALWAYS RESPONDS FALSE
  CODE             local:-1
}
```

6) **Paragraphe 10.5.2**

Le paragraphe 10.5.2 devient le suivant (les adjonctions étant soulignées):

«**10.5.2** L'opération est idempotente et ne renvoie aucune notification en retour.»

7) **Paragraphe 10.6 à 10.16**

Renommer les paragraphes 10.6 à 10.16 en 10.12 à 10.22.

8) **Paragraphe 10.6 à 10.11**

Ajouter les nouveaux paragraphes suivants numérotés de 10.6 à 10.11:

10.6 **Opération probe**

10.6.1 L'opération `probe` (*sondage*) recherche le résultat d'une opération précédemment invoquée. Elle est spécifiée comme suit:

```
probe OPERATION ::=
{
  ARGUMENT SEQUENCE
  {
    invokeId [0] InvokeId
  }
  RESULT      ENUMERATED{running(0), finished(1), unknown(2), ...}
  IDEMPOTENT  TRUE
  CODE        local:-2
}
```

10.6.2 Un seul argument, de type `InvokeId` (*identificateur d'invocation*), identifie l'opération invoquée sur laquelle est effectuée la recherche.

10.6.3 La demande renvoie toujours un résultat, qui indique si l'invocation de l'opération est toujours en cours `running`, si l'exécution de l'invocation est terminée `finished` ou si cette invocation est inconnue `unknown`.

NOTE – Une invocation peut être inconnue parce qu'elle n'a jamais eu lieu ou parce que l'exécutant l'a oubliée.

10.6.4 L'opération est idempotente.

10.6.5 L'opération `probe` (avec le résultat `finished`) provoque, comme effet secondaire, la retransmission de tout retour faisant suite à l'invocation concernée, sauf si l'opération était idempotente.

NOTE – Cela signifie que l'exécutant d'une opération non idempotente doit conserver la réponse (résultat ou erreur) si l'opération `probe` a été incluse dans le lot d'opérations.

10.7 Opération `acknowledge`

10.7.1 L'opération `acknowledge` (*acquiescement*) accuse réception du retour de l'invocation d'une certaine opération (non idempotente). Elle est spécifiée comme suit:

```

acknowledge OPERATION ::=
{
    ARGUMENT    InvokeId
    RESULT      ENUMERATED{acknowledged(0), unknown(1), ...}
    IDEMPOTENT  TRUE
    CODE        local:-3
}

```

10.7.2 Un seul argument, de type `InvokeId` (*identificateur d'invocation*), identifie l'invocation dont le retour est en cours d'acquiescement.

10.7.3 La demande renvoie toujours un résultat, qui indique soit que le retour est maintenant considéré comme acquiescé `acknowledged` soit que l'invocation de l'opération concernée est inconnue `unknown`.

NOTE – Une invocation peut être inconnue parce qu'elle n'a jamais eu lieu ou parce que l'exécutant l'a oubliée.

10.7.4 L'opération est idempotente.

10.7.5 Cette opération doit figurer dans chaque lot d'opérations qui comporte l'opération `probe` (*sondage*).

10.8 Ensemble `Probe and Acknowledge`

10.8.1 L'ensemble d'opérations `ProbeAndAcknowledge` (*sondage et acquiescement*) comprend les deux opérations indiquées par son nom qui seront fréquemment nécessaires ensemble dans les lots. Cet ensemble est spécifié comme suit:

```

ProbeAndAcknowledge OPERATION ::= {probe | acknowledge}

```

10.9 Opération `cancel`

10.9.1 L'opération `cancel` (*annulation*) demande l'arrêt prématuré de l'exécution d'une opération. Seules les opérations qui comportent l'erreur `cancelled` (*annulé*) (voir 10.11) dans leur champ `&Errors` (*erreurs*) peuvent être annulées. Elle est spécifiée comme suit:

```

cancel OPERATION ::=
{
    ARGUMENT    InvokeId
    ERRORS      {cancelFailed}
    IDEMPOTENT  TRUE
    CODE        local:-4
}

```

10.9.2 Un seul argument, de type `InvokeId` (*identificateur d'invocation*), identifie l'opération invoquée qui est en cours d'annulation.

10.9.3 Si la demande échoue, une erreur `cancelFailed` (*échec de l'annulation*) (voir 10.10) sera renvoyée.

10.9.4 L'opération est idempotente.

10.10 Erreur `cancelFailed`

10.10.1 L'erreur `cancelFailed` (*échec de l'annulation*) signale un problème dans l'exécution d'une opération `cancel` (*annulation*). Elle est spécifiée comme suit:

<pre>cancelFailed ERROR ::= { PARAMETER SET { problem [0] CancelProblem, operation [1] InvokeId } CODE local:-2 } CancelProblem ::= ENUMERATED {unknownOperation(0), tooLate(1), operationNotCancellable(2), ...}</pre>

10.10.2 Les divers paramètres ont la signification définie aux 10.10.2.1 et 10.10.2.2.

10.10.2.1 Le problème `problem` particulier rencontré lors de l'annulation est indiqué parmi les possibilités suivantes:

- `unknownOperation` (*opération inconnue*) – L'invocation de cette opération n'a jamais eu lieu, ou elle a été oubliée;
- `tooLate` (*trop tard*) – L'opération a déjà été exécutée, ou l'exécution est à un stade qui ne permet pas l'annulation;
- `operationNotCancellable` (*opération ne pouvant pas être annulée*) – L'opération invoquée ne faisait pas partie de celles qui pouvaient être annulées.

10.10.2.2 Le paramètre `operation` identifie l'opération (invocation) qui devait être annulée.

10.11 Erreur `cancelled`

L'erreur `cancelled` (*annulé*) est signalée lorsqu'une certaine opération est annulée. Elle doit figurer dans le champ `&Errors` de l'opération affectée. Elle est spécifiée comme suit:

<pre>cancelled ERROR ::= {CODE local:-3}</pre>
--

9) Annexe A

Modifier la référence du premier module comme suit (le texte modifié étant souligné):

Remote-Operations-Information-Objects {joint-iso-itu-t remote-operations(4) informationObjects(5) version2(1)}

Ajouter le champ suivant (souligné) à la classe d'objets informationnels OPERATION:

OPERATION ::= CLASS		
{		
	&ArgumentType	OPTIONAL,
	&argumentTypeOptional	BOOLEAN OPTIONAL,
	&returnResult	BOOLEAN DEFAULT TRUE,
	&ResultType	OPTIONAL,
	&resultTypeOptional	BOOLEAN OPTIONAL,
	&Errors	ERROR OPTIONAL,
	&Linked	OPERATION OPTIONAL,
	&synchronous	BOOLEAN DEFAULT FALSE,
	<u>&idempotent</u>	<u>BOOLEAN DEFAULT FALSE,</u>
	&alwaysReturns	BOOLEAN DEFAULT TRUE,
	&InvokePriority	Priority OPTIONAL,
	&ResultPriority	Priority OPTIONAL,
	&operationCode	Code UNIQUE OPTIONAL
}		
WITH SYNTAX		
{		
	[ARGUMENT	&ArgumentType [OPTIONAL
	[RETURN RESULT	&returnResult]
	[RESULT	&ResultType [OPTIONAL
	[ERRORS	&Errors]
	[LINKED	&Linked]
	[SYNCHRONOUS	&synchronous]
	<u>IDEMPOTENT</u>	<u>&idempotent]</u>
	[ALWAYS RESPONDS	&alwaysReturns]
	[INVOKE PRIORITY	&InvokePriority]
	[RESULT-PRIORITY	&ResultPriority]
	[CODE	&operationCode]
}		

Modifier la référence du troisième module comme suit (la modification est soulignée):

Remote-Operations-Useful-Definitions {joint-iso-itu-t remote-operations(4) useful-definitions(7) version2(1)}

Ajouter le champ (souligné) comme suit à la définition de l'opération no-op:

no-op OPERATION ::=	
{	
	<u>IDEMPOTENT</u>
	ALWAYS RESPONDS
	CODE
	<u>TRUE</u>
	FALSE
	local:-1
}	

Ajouter les nouveaux éléments suivants à ce module:

```

probe OPERATION ::=
{
    ARGUMENT SEQUENCE
    {
        invokeId [0] InvokeId
    }
    RESULT ENUMERATED{running(0), finished(1), unknown(2), ...}
    IDEMPOTENT TRUE
    CODE local:-2
}

acknowledge OPERATION ::=
{
    ARGUMENT InvokeId
    RESULT ENUMERATED{acknowledged(0), unknown(1), ...}
    IDEMPOTENT TRUE
    CODE local:-3
}

ProbeAndAcknowledge OPERATION ::= {probe | acknowledge}

cancel OPERATION ::=
{
    ARGUMENT InvokeId
    ERRORS {cancelFailed}
    IDEMPOTENT TRUE
    CODE local:-4
}

cancelFailed ERROR ::=
{
    PARAMETER SET
    {
        problem [0] CancelProblem,
        operation [1] InvokeId
    }
    CODE local:-2
}

CancelProblem ::= ENUMERATED
    {unknownOperation(0), tooLate(1), operationNotCancellable(2), ...}

cancelled ERROR ::= {CODE local:-3}

```

10) Annexe D

Apporter les modifications suivantes au tableau (la modification est soulignée):

Article	Valeur d'identificateur d'objet
Annexe A	{joint-iso-itu-t remote-operations(4) informationObjects(5) version2(1)} {joint-iso-itu-t remote-operations(4) useful-definitions(7) version2(1)}