UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T Q.822.1

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Amendement 1
(03/2003)

SÉRIE Q: COMMUTATION ET SIGNALISATION

Interface Q3

Service RGT de gestion de la qualité de fonctionnement en architecture CORBA

Amendement 1: Gestion des performances du transport générique

Recommandation UIT-T Q.822.1 (2001) – Amendement 1

RECOMMANDATIONS UIT-T DE LA SÉRIE Q

COMMUTATION ET SIGNALISATION

SIGNALISATION DANS LE SERVICE MANUEL INTERNATIONAL	Q.1–Q.3
EXPLOITATION INTERNATIONALE AUTOMATIQUE ET SEMI-AUTOMATIQUE	Q.4–Q.59
FONCTIONS ET FLUX D'INFORMATION DES SERVICES DU RNIS	Q.60–Q.99
CLAUSES APPLICABLES AUX SYSTÈMES NORMALISÉS DE L'UIT-T	Q.100–Q.119
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 4	Q.120–Q.139
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 5	Q.140–Q.199
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 6	Q.250–Q.309
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION R1	Q.310–Q.399
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION R2	Q.400–Q.499
COMMUTATEURS NUMÉRIQUES	Q.500–Q.599
INTERFONCTIONNEMENT DES SYSTÈMES DE SIGNALISATION	Q.600–Q.699
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 7	Q.700–Q.799
INTERFACE Q3	**Q.800–Q.849**
SYSTÈME DE SIGNALISATION D'ABONNÉ NUMÉRIQUE N° 1	Q.850–Q.999
RÉSEAUX MOBILES TERRESTRES PUBLICS	Q.1000–Q.1099
INTERFONCTIONNEMENT AVEC LES SYSTÈMES MOBILES À SATELLITES	Q.1100–Q.1199
RÉSEAU INTELLIGENT	Q.1200–Q.1699
PRESCRIPTIONS ET PROTOCOLES DE SIGNALISATION POUR LES IMT-2000	Q.1700–Q.1799
SPÉÇIFICATIONS DE LA SIGNALISATION RELATIVE À LA COMMANDE D'APPEL INDÉPENDANTE DU SUPPORT	Q.1900–Q.1999
RNIS À LARGE BANDE	Q.2000–Q.2999

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

Recommandation UIT-T Q.822.1

Service RGT de gestion de la qualité de fonctionnement en architecture CORBA

Amendement 1

Gestion des performances du transport générique

Résumé

Le présent amendement définit une nouvelle classe d'objets à utiliser pour représenter les données de gestion de la performance (PM, *performance management*) d'un accès physique ou d'extrémités de connexions de transport. Cette nouvelle classe d'objets de gestion des performances du transport générique (*generic transport PM*) est censée être applicable à différentes technologies, différentes architectures et différents services. Le modèle de langage de définition d'interface (IDL, *interface definition language*) décrit la nouvelle classe d'objets GenericTransportPmCD et toutes les interfaces qui lui sont associées.

Source

L'Amendement 1 de la Recommandation Q.822.1 (2001) de l'UIT-T, élaboré par la Commission d'études 4 (2001-2004) de l'UIT-T, a été approuvé le 29 mars 2003 selon la procédure définie dans la Résolution 1 de l'AMNT.

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'UIT. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

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

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

Dans certains secteurs des technologies de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

NOTE

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

DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

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

TABLE DES MATIÈRES

Recommandation UIT-T Q.822-1

Service RGT de gestion de la qualité de fonctionnement en architecture CORBA

Amendement 1

Gestion des performances du transport générique

1 Domaine d'application

Le présent amendement définit une nouvelle classe d'objets à utiliser pour représenter les données de gestion de la performance (PM, *performance management*) d'un accès physique ou d'extrémités de connexions de transport. Cette nouvelle classe d'objets de gestion des performances du transport générique (*generic transport PM*) est censée être applicable à différentes technologies, différentes architectures et différents services. Le modèle de langage de définition d'interface (IDL, *interface definition language*) décrit la nouvelle classe d'objets GenericTransportPmCD et toutes les interfaces qui lui sont associées.

2 Références normatives

La présente Recommandation se réfère à certaines dispositions des Recommandations UIT-T et textes suivants qui, de ce fait, en sont partie intégrante. Les versions indiquées étaient en vigueur au moment de la publication de la présente Recommandation. Toute Recommandation ou tout texte étant sujet à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des références normatives suivantes. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée. La référence à un document figurant dans la présente Recommandation ne donne pas à ce document, en tant que tel, le statut d'une Recommandation.

[1] Recommandation UIT-T Q.816 (2001), *Services RGT à architecture CORBA*.

[2] Recommandation UIT-T Q.816.1 (2001), *Services RGT à architecture CORBA: extensions pour la prise en charge des interfaces à granularité grossière*.

[3] Recommandation UIT-T X.780 (2001), *Directives concernant le RGT pour la définition d'objets gérés CORBA*.

[4] Recommandation UIT-T X.780.1 (2001), *Directives concernant le RGT pour la définition d'interfaces d'objets gérés CORBA à granularité grossière*.

[5] Recommandation UIT-T M.3100 (1995), *Modèle générique d'information de réseau*, plus Amendement 2 (2000).

[6] Recommandation UIT-T Q.822.1 (2001), *Service RGT de gestion de la qualité de fonctionnement en architecture CORBA*.

[7] ANSI Standard T1.231-1997, *Digital Hierarchy – Layer 1 in-Service Digital Transmission Performance Monitoring*.

3 Aperçu général du modèle informationnel applicable aux données de gestion des performances du transport générique

Le présent amendement définit une nouvelle classe d'objets de données de gestion des performances du transport générique. Cette nouvelle classe d'objets est utilisée pour recueillir des données de

performance relatives à l'attribut GenericTransportTTP (la terminaison d'une connexion de transport générique). Les classes GenericTransportPmCDValueType et GenericTransportPmHDValueType contiennent des emplacements réservés pour les valeurs des données actuelles et des données chronologiques du groupe de compteurs assurant la surveillance du protocole de transport physique.

La classe d'objets GenericTransportPmCD est une sous-classe de la classe d'objets CurrentData définie dans les Recommandations UIT-T Q.822 et Q.822.1. Les méthodes héritées de la classe CurrentData permettent d'extraire les attributs SuspectFlag et ElapsedTime, d'activer ou de désactiver l'attribut HistoryRetention, d'associer des données de seuil à une instance de données actuelles, et d'activer ou de désactiver la suppression des compteurs n'affichant que des zéros. Les autres méthodes héritées du releveur (scanner) permettent de définir l'attribut AdministrativeState, d'extraire l'attribut OperationalState et de définir la période de granularité. De même, la classe GenericTransportPmHD est une sous-classe de la classe HistoryData.

Pour certains signaux numériques, les primitives de performance dans la direction d'arrivée sont communiquées à l'extrémité distante au moyen de messages spéciaux intégrés dans le format des signaux, tels que, par exemple, le message de signalisation de la performance (PRM, *performance report message*) intégré dans le canal DS1 ESF, les indicateurs d'erreur de bloc à l'extrémité distante (FEBE, *far-end block error*) intégrés dans les applications DS3 CC-bit ou les indicateurs d'erreur distante (REI, *remote error indicators*) utilisés en hiérarchie numérique synchrone (SDH). Une telle capacité intégrée dans un signal de transmission permet de communiquer à l'extrémité proche les paramètres de qualité de transmission observés à l'extrémité distante. C'est la raison pour laquelle l'entité gérée chargée de conserver les données de gestion de performance du point TTP de transport générique contient des emplacements réservés pour stocker les données de l'extrémité proche et celles de l'extrémité distante.

Les attributs de l'objet GenericTransportPmCD sont répartis dans six paquetages, respectivement dénommés: chemin d'accès à l'extrémité proche, ligne d'extrémité proche, section d'extrémité proche, chemin d'accès à l'extrémité distante, ligne d'extrémité distante et section d'extrémité distante. Seuls les attributs du paquetage chemin d'accès à l'extrémité proche sont obligatoires. D'autres paquetages peuvent être utilisés chaque fois qu'on le jugera utile compte tenu de la technologie de transport sous-jacente donnant lieu à modélisation.

Le paragraphe 4 définit un ensemble d'interfaces IDL CORBA pour les objets GenericTransportPmCD et GenericTransportPmHD. Ces interfaces sont spécifiées selon le cadre CORBA du RGT et les directives données dans les Recommandations UIT-T Q.816 et X.780 pour l'interface CORBA à granularité fine.

Outre les interfaces à granularité fine définies au § 4.4, un ensemble d'interfaces-façades associées est défini au § 4.5. Ces interfaces-façades sont définies conformément au cadre à granularité grossière et aux directives données dans les Recommandations UIT-T Q.816.1 et X.780.1 pour la prise en charge de l'interface CORBA à granularité grossière. Ces interfaces-façades sont désignées par la même dénomination que celle de l'interface à granularité fine correspondante, suivie de "_F" (un trait de soulignement suivi d'un "F" majuscule).

Le langage IDL figurant dans le présent amendement fait partie intégrante de la Rec. UIT-T Q.822.1. Cela suppose que toutes les définitions (classes d'objets, type, structure, etc.) définies dans la Rec. UIT-T Q.822.1 figurent dans le même module IDL et que l'on puisse s'y référer sans connaître l'identificateur du module.

Le langage IDL figurant dans le présent amendement a été compilé avec succès sans erreur de syntaxe. Le compilateur utilisé avait été déclaré conforme au CORBA 2.3, lequel inclut le type de valeur et les capacités macro M4.

Les Figures 1 et 2 indiquent les relations d'héritage, de confinement et d'association des interfaces CORBA définies dans le présent amendement. A noter que les interfaces-façades suivent la même relation de hiérarchie d'héritage que les interfaces à granularité fine correspondantes.

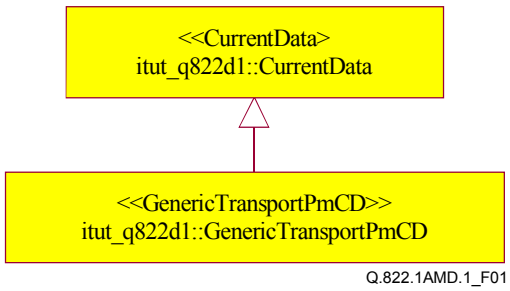

Figure 1/Q.822.1/Amd.1 – Relation d'héritage

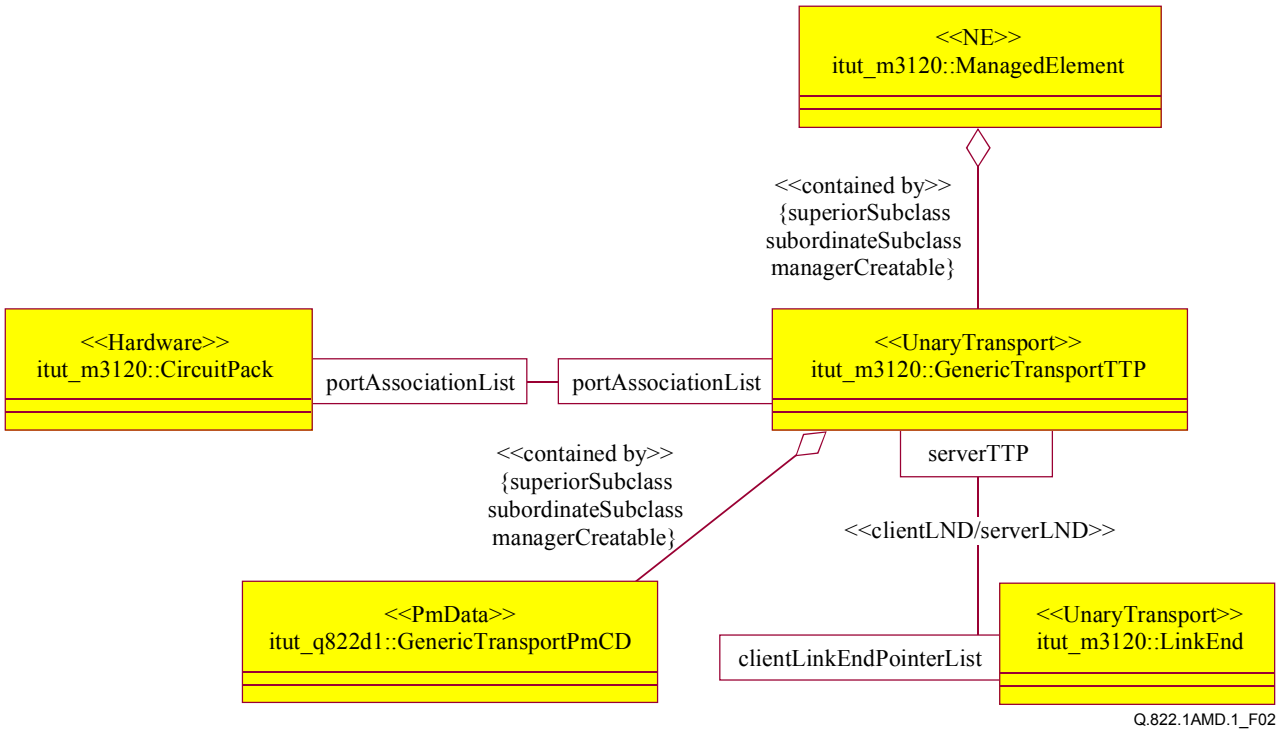

Figure 2/Q.822.1/Amd.1 – Relations de confinement et d'association

4 Langage IDL du modèle d'information

```
#ifndef _itut_q822d1_amd1_idl_
#define _itut_q822d1_amd1_idl_


#include <itut_q822d1.idl>


#pragma prefix "itu.int"


/**
Ce code IDL est destiné à être mémorisé dans un fichier nommé
"itut_q822d1_amd1.idl"
qui est situé dans le chemin de recherche utilisé par les compilateurs IDL de
votre système. Le module principal (défini dans la Rec. UIT-T Q.822.1)
est confiné dans un fichier séparé nommé
"itut_q822d1.idl"
*/
```

```
/**
Ce fragment est ajouté au module, itut_q822d1, qui contient la définition du
langage IDL
basée sur les objets définis dans la Rec. UIT-T Q.822.
*/
module itut_q822d1
{


/**
```

4.1 Déclarations anticipées

```
*/

/**
Déclarations anticipées d'interface
*/
       interface GenericTransportPmCD;

/**
Déclarations anticipées de type de valeur (Valuetype)
*/
     valuetype GenericTransportPmCDValueType;
valuetype GenericTransportTTPHDValueType;


/**
Déclarations anticipées de définitions de type (Typedefs)
*/
     typedef MONameType GenericTransportPmCDNameType;


/**
```

4.2 Structures et définitions de type

```
Aucune nouvelle structure n'est définie dans le présent amendement.
*/

/**
```

4.3 Exceptions applicables aux paquetages conditionnels

```
*/
       exception NOfarEndLinePMDataPackage {};
       exception NOfarEndPathPMDataPackage {};
       exception NOfarEndSectionPMDataPackage {};
       exception NOnearEndLinePMDataPackage {};
       exception NOnearEndSectionPMDataPackage {};



/**
```

4.4 Interfaces à granularité fine

```
*/

/**
```

4.4.1 GenericTransportPmCD (Données actuelles de surveillance de la performance du transport physique)

```
Cet objet permet d'extraire les attributs ou les valeurs des données actuelles
et des données chronologiques du groupe de compteurs assurant la surveillance du
protocole de transport physique.

Les autres méthodes héritées du releveur (scanner) permettent de définir
l'attribut AdministrativeState, d'extraire l'attribut OperationalState et de
définir la période de granularité. Les méthodes héritées de la classe
CurrentData permettent d'extraire les attributs SuspectFlag et ElapsedTime,
d'activer ou de désactiver l'attribut HistoryRetention, d'associer des données
de seuil à une instance de données actuelles, et d'activer ou de désactiver la
suppression des compteurs n'affichant que des zéros.

Les méthodes applicables héritées du releveur comprennent les éléments suivants:
administrativeStateGet, administrativeStateSet, operationalStateGet,
granularityPeriodGet, et granularityPeriodSet.

Les méthodes applicables héritées de la classe CurrentData comprennent les
éléments suivants: suspectIntervalFlagGet, suspectIntervalFlagDefaultSet,
elapsedTimeGet, historyRetentionGet, historyRetentionSet,
thresholdDataInstanceListGet, thresholdDataInstanceListSet, getMostRecent,
getBetween, numSuppressedIntervalsGet, maxSuppressedIntervalsGet,
maxSuppressedIntervalsSet.

Cet objet doit prendre en charge les notifications suivantes: objectCreation,
objectDeletion, attributeValueChange, et stateChange.

Lorsque l'un quelconque des paramètres de performance dépasse une valeur de
seuil préétablie indiquée dans l'objet ThresholdData associé, le système de
gestion en est informé par une alerte de dépassement de seuil ou alarme de
qualité de service (qualityOfServiceAlarm).

Il existe une relation de confinement entre un attribut GenericTransportTTP et
l'objet GenericTransportPmCD qui lui est associé.
*/

valuetype GenericTransportPmCDValueType: truncatable
itut_q822d1::CurrentDataValueType
     {

/**
On utilise le paramètre codingViolationsPath pour compter certains éléments
d'erreur se produisant pendant la période d'accumulation. Comme exemples
d'évènements d'erreur citons les erreurs de synchronisation de trames et de
contrôle de redondance cyclique (CRC) pour les liaisons DS1, ou les erreurs de
parité sur les bits P ou CP pour les liaisons DS3.
*/
               public unsigned long codingViolationsPath;
                       // GET
/**
Le paramètre erroredSecondsPath permet de compter les intervalles de 1 seconde
comportant des erreurs de chemin d'accès. Comme exemples d'erreurs de chemin
d'accès, citons les erreurs CRC-6 (DS1), les défauts dus à des trames comportant
beaucoup d'erreurs (DS1/DS3) et les erreurs de parité sur le bit P (DS3).
*/
               public unsigned long erroredSecondsPath;
                       // GET
/**
Le paramètre severelyErroredSecondspath permet de compter les intervalles de
1 seconde comportant X évènements d'erreur de chemin d'accès ou plus, ou un ou
plusieurs défauts de perte de signal.
```

```
*/
                public unsigned long severelyErroredSecondsPath;
                        // GET
/**
Le paramètre unavailableSecondsPath permet de compter les intervalles de
1 seconde pendant lesquels le chemin d'accès est indisponible.
*/
                public unsigned long unavailableSecondsPath;
                        // GET
                public unsigned long failureCounterPath;
                        // GET
                public unsigned long farEndCodingViolationsPath;
                        // GET
                        // farEndPathPMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndErroredSecondsPath;
                        // GET
                        // farEndPathPMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndSeverelyErroredSecondsPath;
                        // GET
                        // farEndPathPMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndUnavailableSecondsPath;
                        // GET
                        // farEndPathPMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndFailureCounterPath;
                        // GET
                        // farEndPathPMDataPackage
                        // Conditional, present if the instance supports it
/**
Le paramètre codingViolationsLine permet de compter certains évènements d'erreur
se produisant pendant la période d'accumulation. Comme exemples d'évènements
d'erreur, citons les violations bipolaires (BPV, bipolar violations) et
l'apparition d'un nombre excessif de zéros (EXZ, excessive zeros) sur une
liaison DS1/DS3.
*/
                public unsigned long codingViolationsLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
/**
Le paramètre erroredSecondsLine permet de compter les intervalles de 1 seconde
pendant lesquels un ou plusieurs évènements d'erreur de violation du codage en
ligne ont été détectés.
*/
                public unsigned long erroredSecondsLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
/**
Le paramètre severelyErroredSecondsLine permet de compter les intervalles de
1 seconde comportant X violations BPV ou plus et l'apparition d'un nombre
excessif de zéros (EXZ), ou un ou plusieurs défauts de perte de signal.
*/
                public unsigned long severelyErroredSecondsLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
/**
Le paramètre lossOfSignalLine permet de compter les intervalles de 1 seconde
comportant un ou plusieurs défauts de perte de signal.
```

```
*/
                public unsigned long lossOfSignalLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long failureCounterLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndCodingViolationsLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndErroredSecondsLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndSeverelyErroredSecondsLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndlossOfSignalLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndFailureCounterLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long codingViolationsSection;
                        // GET
                        // nearEndSectionPMDataPackage
                        // Conditional, present if the instance supports sections

                public unsigned long erroredSecondsSection;
                        // GET
                        // nearEndSectionPMDataPackage
                        // Conditional, present if the instance supports sections

                public unsigned long severelyErroredSecondsSection;
                        // GET
                        // nearEndSectionPMDataPackage
                        // Conditional, present if the instance supports sections

                public unsigned long lossOfSignalSection;
                        // GET
                        // nearEndSectionPMDataPackage
                        // Conditional, present if the instance supports sections

                public unsigned long failureCounterSection;
                        // GET
                        // nearEndSectionPMDataPackage
                        // Conditional, present if the instance supports sections

                public unsigned long farEndCodingViolationsSection;
                        // GET
                        // farEndSectionPMDataPackage
                        // Conditional, present if the instance supports sections

                public unsigned long farEndErroredSecondsSection;
                        // GET
                        // farEndSectionPMDataPackage
                        // Conditional, present if the instance supports sections
```

```
                public unsigned long farEndSeverelyErroredSecondsSection;
                        // GET
                        // farEndSectionPMDataPackage
                        // Conditional, present if the instance supports sections

                public unsigned long farEndlossOfSignalSection;
                        // GET
                        // farEndSectionPMDataPackage
                        // Conditional, present if the instance supports sections

                public unsigned long farEndFailureCounterSection;
                        // GET
                        // farEndSectionPMDataPackage
                        // Conditional, present if the instance supports sections

     }; // valuetype GenericTransportPmCDValueType



valuetype GenericTransportTTPHDValueType: truncatable
itut_q822d1::HistoryDataValueType
     {

/**
Le paramètre codingViolationsPath sert à compter certains évènements d'erreur se
produisant pendant la période d'accumulation. Comme exemples d'évènements
d'erreur, citons les erreurs de synchronisation de trame et les erreurs CRC pour
des liaisons DS1, ou les erreurs de parité sur les bits P ou CP pour les
liaisons DS3.
*/
                public unsigned long codingViolationsPath;
                        // GET
/**
Le paramètre erroredSecondsPath permet de compter les intervalles de 1 seconde
comportant des erreurs de chemin d'accès. Comme exemples d'erreurs de chemin
d'accès, citons les erreurs CRC-6 (DS1), les défauts dus à des trames comportant
beaucoup d'erreurs (DS1/DS3) et les erreurs de parité sur le bit P (DS3).
*/
                public unsigned long erroredSecondsPath;
                        // GET
/**
Le paramètre severelyErroredSecondspath permet de compter les intervalles de 1
seconde comportant X évènements d'erreur de chemin d'accès ou plus, ou un ou
plusieurs défauts de perte de signal.
*/
                public unsigned long severelyErroredSecondsPath;
                        // GET
/**
Le paramètre unavailableSecondsPath permet de compter les intervalles de 1
seconde pendant lesquels le chemin d'accès est indisponible.
*/
                public unsigned long unavailableSecondsPath;
                        // GET
                public unsigned long failureCounterPath;
                        // GET
                public unsigned long farEndCodingViolationsPath;
                        // GET
                        // farEndPathPMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndErroredSecondsPath;
                        // GET
                        // farEndPathPMDataPackage
                        // Conditional, present if the instance supports it
```

```
                public unsigned long farEndSeverelyErroredSecondsPath;
                        // GET
                        // farEndPathPMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndUnavailableSecondsPath;
                        // GET
                        // farEndPathPMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndFailureCounterPath;
                        // GET
                        // farEndPathPMDataPackage
                        // Conditional, present if the instance supports it
/**
Le paramètre codingViolationsLine permet de compter certains évènements d'erreur
se produisant pendant la période d'accumulation. Comme exemples d'évènements
d'erreur, citons les violations bipolaires (BPV) et l'apparition d'un nombre
excessif de zéros (EXZs)sur une liaison DS1/DS3.
*/
                public unsigned long codingViolationsLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
/**
Le paramètre erroredSecondsLine permet de compter les intervalles de 1 seconde
pendant lesquels un ou plusieurs évènements d'erreur de violation de codage en
ligne ont été détectés.
*/
                public unsigned long erroredSecondsLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
/**
Le paramètre severelyErroredSecondsLine permet de compter les intervalles de 1
seconde comportant X violations BPVs ou plus et l'apparition d'un nombre
excessif de zéros (EXZ), ou un ou plusieurs défauts de perte de signal.
*/
                public unsigned long severelyErroredSecondsLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
/**
Le paramètre lossOfSignalLine parameter permet de compter les intervalles de 1
seconde comportant un ou plusieurs défauts de perte de signal.
*/
                public unsigned long lossOfSignalLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long failureCounterLine;
                        // GET
                        // nearEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndCodingViolationsLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndErroredSecondsLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
                public unsigned long farEndSeverelyErroredSecondsLine;
                        // GET
                        // farEndLinePMDataPackage
                        // Conditional, present if the instance supports it
```

```
        public unsigned long farEndlossOfSignalLine;
                // GET
                // farEndLinePMDataPackage
                // Conditional, present if the instance supports it
        public unsigned long farEndFailureCounterLine;
                // GET
                // farEndLinePMDataPackage
                // Conditional, present if the instance supports it
        public unsigned long codingViolationsSection;
                // GET
                // nearEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long erroredSecondsSection;
                // GET
                // nearEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long severelyErroredSecondsSection;
                // GET
                // nearEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long lossOfSignalSection;
                // GET
                // nearEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long failureCounterSection;
                // GET
                // nearEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long farEndCodingViolationsSection;
                // GET
                // farEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long farEndErroredSecondsSection;
                // GET
                // farEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long farEndSeverelyErroredSecondsSection;
                // GET
                // farEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long farEndlossOfSignalSection;
                // GET
                // farEndSectionPMDataPackage
                // Conditional, present if the instance supports sections

        public unsigned long farEndFailureCounterSection;
                // GET
                // farEndSectionPMDataPackage
                // Conditional, present if the instance supports sections


};  // valuetype GenericTransportTTPHDValueType

interface GenericTransportPmCD: itut_q822d1::CurrentData
{
```

```
unsigned long codingViolationsPathGet()
        raises (itut_x780::ApplicationError);

unsigned long erroredSecondsPathGet()
        raises (itut_x780::ApplicationError);

unsigned long severelyErroredSecondsPathGet()
        raises (itut_x780::ApplicationError);

unsigned long unavailableSecondsPathGet()
        raises (itut_x780::ApplicationError);

unsigned long failureCounterPathGet()
        raises (itut_x780::ApplicationError);

unsigned long farEndCodingViolationsPathGet()
        raises (itut_x780::ApplicationError,
                NOfarEndPathPMDataPackage);

unsigned long farEndErroredSecondsPathGet()
        raises (itut_x780::ApplicationError,
                NOfarEndPathPMDataPackage);

unsigned long farEndSeverelyErroredSecondsPathGet()
        raises (itut_x780::ApplicationError,
                NOfarEndPathPMDataPackage);

unsigned long farEndUnavailableSecondsPathGet()
        raises (itut_x780::ApplicationError,
                NOfarEndPathPMDataPackage);

unsigned long farEndFailureCounterPathGet()
        raises (itut_x780::ApplicationError,
                NOfarEndPathPMDataPackage);

unsigned long codingViolationsLineGet()
        raises (itut_x780::ApplicationError,
                NOnearEndLinePMDataPackage);

unsigned long erroredSecondsLineGet()
        raises (itut_x780::ApplicationError,
                NOnearEndLinePMDataPackage);

unsigned long severelyErroredSecondsLineGet()
        raises (itut_x780::ApplicationError,
                NOnearEndLinePMDataPackage);

unsigned long lossOfSignalLineGet()
        raises (itut_x780::ApplicationError,
                NOnearEndLinePMDataPackage);

unsigned long failureCounterLineGet()
        raises (itut_x780::ApplicationError,
                NOnearEndLinePMDataPackage);

unsigned long farEndCodingViolationsLineGet()
        raises (itut_x780::ApplicationError,
                NOfarEndLinePMDataPackage);

unsigned long farEndErroredSecondsLineGet()
        raises (itut_x780::ApplicationError,
                NOfarEndLinePMDataPackage);
```

```
        unsigned long farEndSeverelyErroredSecondsLineGet()
                raises (itut_x780::ApplicationError,
                        NOfarEndLinePMDataPackage);

        unsigned long farEndlossOfSignalLineGet()
                raises (itut_x780::ApplicationError,
                        NOfarEndLinePMDataPackage);

        unsigned long farEndFailureCounterLineGet()
                raises (itut_x780::ApplicationError,
                        NOfarEndLinePMDataPackage);

        unsigned long codingViolationsSectionGet()
                raises (itut_x780::ApplicationError,
                        NOnearEndSectionPMDataPackage);

        unsigned long erroredSecondsSectionGet()
                raises (itut_x780::ApplicationError,
                        NOnearEndSectionPMDataPackage);

        unsigned long severelyErroredSecondsSectionGet()
                raises (itut_x780::ApplicationError,
                        NOnearEndSectionPMDataPackage);

        unsigned long lossOfSignalSectionGet()
                raises (itut_x780::ApplicationError,
                        NOnearEndSectionPMDataPackage);

        unsigned long failureCounterSectionGet()
                raises (itut_x780::ApplicationError,
                        NOnearEndSectionPMDataPackage);

        unsigned long farEndCodingViolationsSectionGet()
                raises (itut_x780::ApplicationError,
                        NOfarEndSectionPMDataPackage);

        unsigned long farEndErroredSecondsSectionGet()
                raises (itut_x780::ApplicationError,
                        NOfarEndSectionPMDataPackage);

        unsigned long farEndSeverelyErroredSecondsSectionGet()
                raises (itut_x780::ApplicationError,
                        NOfarEndSectionPMDataPackage);

        unsigned long farEndlossOfSignalSectionGet()
                raises (itut_x780::ApplicationError,
                        NOfarEndSectionPMDataPackage);

        unsigned long farEndFailureCounterSectionGet()
                raises (itut_x780::ApplicationError,
                        NOfarEndSectionPMDataPackage);

};  // interface GenericTransportPmCD


interface GenericTransportPmCDFactory: itut_x780::ManagedObjectFactory
{
     itut_x780::ManagedObject create
           (in NameBindingType nameBinding,
                 // module name containing Name Binding info.
            in MONameType superior,
                 // Name of containing object.
```

```
                in string reqID,
                      // Requested ID value for name, will be
                      // empty if auto-naming is to be used.
                out MONameType name,
                      // Entire name of newly created object.
                in StringSetType packageNameList,
                      // List of packages requested.

                in short historyRetention,
                in AdministrativeStateType administrativeState,
                in TimePeriodType granularityPeriod,
                in MONameSetType thresholdDataInstanceList,
                in short maxSuppressedIntervals,
                in GeneralizedTimeType periodSynchronizationTime)
                raises (itut_x780::ApplicationError,
                      itut_x780::CreateError);

    }; // interface GenericTransportPmCDFactory

/**
```

4.5 Interfaces-façades

Le comportement des interfaces-façades est identique à celui des interfaces à granularité fine correspondantes. C'est pourquoi aucune observation n'est incluse dans les interfaces-façades. Pour le comportement de l'interface-façade, nous invitons le lecteur à se reporter aux interfaces à granularité fine définies au § 4.4.

Il n'y a pas lieu d'inclure le présent paragraphe dans le langage IDL dans le cas où un système de gestion admet uniquement les interfaces à granularité fine.

```
*/


/**
```

4.5.1 GenericTransportPmCD_F

```
*/
    interface GenericTransportPmCD_F: itut_q822d1::CurrentData_F
    {

            unsigned long codingViolationsPathGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError);

            unsigned long erroredSecondsPathGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError);

            unsigned long severelyErroredSecondsPathGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError);

            unsigned long unavailableSecondsPathGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError);

            unsigned long failureCounterPathGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError);
```

```
unsigned long farEndCodingViolationsPathGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOfarEndPathPMDataPackage);

unsigned long farEndErroredSecondsPathGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOfarEndPathPMDataPackage);

unsigned long farEndSeverelyErroredSecondsPathGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOfarEndPathPMDataPackage);

unsigned long farEndUnavailableSecondsPathGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOfarEndPathPMDataPackage);

unsigned long farEndFailureCounterPathGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOfarEndPathPMDataPackage);

unsigned long codingViolationsLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOnearEndLinePMDataPackage);

unsigned long erroredSecondsLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOnearEndLinePMDataPackage);

unsigned long severelyErroredSecondsLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOnearEndLinePMDataPackage);

unsigned long lossOfSignalLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOnearEndLinePMDataPackage);

unsigned long failureCounterLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOnearEndLinePMDataPackage);

unsigned long farEndCodingViolationsLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOfarEndLinePMDataPackage);

unsigned long farEndErroredSecondsLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOfarEndLinePMDataPackage);

unsigned long farEndSeverelyErroredSecondsLineGet
          (in MONameType name)
          raises (itut_x780::ApplicationError,
                 NOfarEndLinePMDataPackage);
```

```
            unsigned long farEndlossOfSignalLineGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError,
                             NOfarEndLinePMDataPackage);

            unsigned long farEndFailureCounterLineGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError,
                             NOfarEndLinePMDataPackage);

            unsigned long codingViolationsSectionGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError,
                             NOnearEndSectionPMDataPackage);

            unsigned long erroredSecondsSectionGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError,
                             NOnearEndSectionPMDataPackage);

            unsigned long severelyErroredSecondsSectionGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError,
                             NOnearEndSectionPMDataPackage);

            unsigned long lossOfSignalSectionGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError,
                             NOnearEndSectionPMDataPackage);

            unsigned long failureCounterSectionGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError,
                             NOnearEndSectionPMDataPackage);

            unsigned long farEndCodingViolationsSectionGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError,
                             NOfarEndSectionPMDataPackage);

            unsigned long farEndErroredSecondsSectionGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError,
                             NOfarEndSectionPMDataPackage);

            unsigned long farEndSeverelyErroredSecondsSectionGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError,
                             NOfarEndSectionPMDataPackage);

            unsigned long farEndlossOfSignalSectionGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError,
                             NOfarEndSectionPMDataPackage);

            unsigned long farEndFailureCounterSectionGet
                      (in MONameType name)
                      raises (itut_x780::ApplicationError,
                             NOfarEndSectionPMDataPackage);

      }; // interface GenericTransportPmCD_F
/**
```

4.6 Corrélations de noms

```
Le module suivant contient l'information de corrélation de noms.
*/
     module NameBinding
     {

/**
```

4.6.1 GenericTransportPmCD

```
*/
              module GenericTransportPmCD_GenericTransportTTP
              {
                    const string   superiorClass =
                              "itut_m3120:: GenericTransportTTP";
                    const boolean  superiorSubclassesAllowed = TRUE;
                    const string   subordinateClass =
                              "itut_m3120:: GenericTransportPmCD";
                    const boolean  subordinateSubclassesAllowed = TRUE;
                    const boolean  managerCreatesAllowed = TRUE;
                    const DeletePolicyType deletePolicy =
                              itut_x780::deleteOnlyIfNoContainedObjects;
                    const string   kind = "GenericTransportPmCD";
              }; // module GenericTransportPmCD_GenericTransportTTP

     }; // module NameBinding

}; // module itut_q822d1

#endif // _itut_q822d1_amd1_idl_
```

SÉRIES DES RECOMMANDATIONS UIT-T

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

Imprimé en Suisse
Genève, 2003