



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

M.3120

Amendement 2
(03/2003)

SÉRIE M: RGT ET MAINTENANCE DES RÉSEAUX:
SYSTÈMES DE TRANSMISSION, CIRCUITS
TÉLÉPHONIQUES, TÉLÉGRAPHIE, TÉLÉCOPIE ET
CIRCUITS LOUÉS INTERNATIONAUX

Réseau de gestion des télécommunications

Modèle générique informationnel d'architecture
CORBA des réseaux et éléments de réseau

Amendement 2

Recommandation UIT-T M.3120 (2001) – Amendement 2

RECOMMANDATIONS UIT-T DE LA SÉRIE M
**RGT ET MAINTENANCE DES RÉSEAUX: SYSTÈMES DE TRANSMISSION, CIRCUITS
TÉLÉPHONIQUES, TÉLÉGRAPHIE, TÉLÉCOPIE ET CIRCUITS LOUÉS INTERNATIONAUX**

Introduction et principes généraux de maintenance et organisation de la maintenance	M.10–M.299
Systèmes de transmission internationaux	M.300–M.559
Circuits téléphoniques internationaux	M.560–M.759
Systèmes de signalisation à canal sémaphore	M.760–M.799
Systèmes internationaux de télégraphie et de phototélégraphie	M.800–M.899
Liaisons internationales louées par groupes primaires et secondaires	M.900–M.999
Circuits internationaux loués	M.1000–M.1099
Systèmes et services de télécommunication mobile	M.1100–M.1199
Réseau téléphonique public international	M.1200–M.1299
Systèmes internationaux de transmission de données	M.1300–M.1399
Appellations et échange d'informations	M.1400–M.1999
Réseau de transport international	M.2000–M.2999
Réseau de gestion des télécommunications	M.3000–M.3599
Réseaux numériques à intégration de services	M.3600–M.3999
Systèmes de signalisation par canal sémaphore	M.4000–M.4999

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

Recommandation UIT-T M.3120

Modèle générique informationnel d'architecture CORBA des réseaux et éléments de réseau

Amendement 2

Résumé

Le présent amendement propose plusieurs améliorations du modèle générique informationnel d'architecture CORBA des réseaux et éléments de réseau. Tout d'abord, il décrit en détail les mécanismes qui prennent en charge la transmission des gammes de valeurs d'attribut via l'interface CORBA. Ensuite, il définit une nouvelle classe d'objets TTP de transport générique qui a pour but de représenter le port physique ou les points d'extrémité des connexions d'acheminement. Enfin, il définit une nouvelle classe d'objets, ManagedElementR2, qui est une sous-classe de ManagedElement et trois nouveaux attributs. Il s'agit d'un attribut contenant le "code du modèle" d'un équipement, d'un attribut servant à représenter des pseudonymes d'éléments de réseau ou des noms utilisés par l'EMS pour désigner des éléments de réseau, et d'un attribut contenant le "type" générique d'un élément de réseau.

Une autre amélioration qu'apporte le présent amendement concerne l'élargissement du module de constantes CharacteristicInfo afin qu'il puisse représenter le mieux possible les débits de signal actuellement disponibles.

Source

L'Amendement 2 de la Recommandation M.3120 (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.

© UIT 2003

Tous droits réservés. Aucune partie de cette publication ne peut être reproduite, par quelque procédé que ce soit, sans l'accord écrit préalable de l'UIT.

TABLE DES MATIÈRES

	Page
1	Domaine d'application 1
2	Références normatives..... 1
3	Termes et définitions 2
4	Abréviations et acronymes 2
5	Conventions 2
6	Rappel des gammes de valeurs d'attribut..... 2
7	Aperçu général de la classe d'objets TTP de transport générique 5
8	Amélioration de la classe d'objets ManagedElement 6
8.1	Code du modèle 7
8.2	Pseudonymes des éléments de réseau..... 7
8.3	Type d'élément de réseau 7
9	Développement des informations sur les caractéristiques 7
10	Modèle d'information 8
10.1	Structures et TypeDefs 9
10.2	Interfaces – A granulation fine 11
10.3	Interfaces – Façade 17
10.4	Corrélation de noms 19

Recommandation UIT-T M.3120

Modèle générique informationnel d'architecture CORBA des réseaux et éléments de réseau

Amendement 2

1 Domaine d'application

Le présent amendement propose plusieurs améliorations du modèle générique informationnel d'architecture CORBA des réseaux et éléments de réseau. Tout d'abord, il décrit en détail les mécanismes qui prennent en charge la transmission des gammes de valeurs d'attribut via l'interface CORBA. Ensuite, il définit une nouvelle classe d'objets TTP de transport générique qui a pour but de représenter le port physique ou les points d'extrémité des connexions d'acheminement. Enfin, il définit une nouvelle classe d'objets, ManagedElementR2, qui est une sous-classe de ManagedElement et trois nouveaux attributs. Il s'agit d'un attribut contenant le "code du modèle" d'un équipement, d'un attribut servant à représenter des pseudonyme d'éléments de réseau ou des noms utilisés par l'EMS pour désigner des éléments de réseau, et d'un attribut contenant le "type" générique d'un élément de réseau.

Une autre amélioration qu'apporte le présent amendement concerne l'élargissement du module de constantes CharacteristicInfo afin qu'il puisse représenter le mieux possible les débits de signal actuellement disponibles.

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 des 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.3120 (2001), *Modèle générique informationnel d'architecture CORBA des réseaux et éléments de réseau*.
- [6] Recommandation UIT-T M.3100 (1995), *Modèle générique d'information de réseau plus Amendement 1 (1999)*.

- [7] Recommandation UIT-T Q.822.1 (2001), *Service RGT de gestion de la qualité de fonctionnement en architecture CORBA*.
- [8] Norme ANSI T1.231.1997 (1997), *Digital Hierarchy – Layer 1 in-Service Digital Transmission Performance Monitoring*.

3 Termes et définitions

Le présent amendement ne contient pas de termes autres que ceux utilisés dans la Recommandation de base.

4 Abréviations et acronymes

Le présent amendement ne contient pas d'abréviations autres que celles utilisées dans la Recommandation de base.

5 Conventions

Le présent amendement ne contient pas de conventions autres que celles utilisées dans la Recommandation de base.

6 Rappel des gammes de valeurs d'attribut

Le présent paragraphe propose un mécanisme qui permet à des systèmes gérés utilisant le paradigme M.3120 de signaler automatiquement des gammes de valeurs acceptables pour les attributs associés à un élément de réseau du modèle. Un tel mécanisme pourrait être une ressource très intéressante pour la découverte et la configuration de l'équipement étant donné qu'un système de gestion connaîtrait automatiquement les gammes de valeurs acceptables pour chaque paramètre configurable du réseau avant de tenter d'utiliser ces valeurs.

Pour mettre ce mécanisme en pratique, nous définissons une nouvelle classe d'objets *AttributeRanges* (gammes d'attributs). Celle-ci permet au système géré de signaler les valeurs minimale et maximale acceptées par certains attributs, ainsi que la granularité, ou valeurs de progression, de la gamme. Chaque instance de *AttributeRanges* contient les gammes d'attributs appartenant à une classe d'objets. L'attribut "*kind*" (nature) de *AttributeRanges* désigne la classe d'objets pour laquelle les gammes sont définies, l'attribut "*attributeName*" spécifie le nom de l'attribut pour lequel les gammes sont définies. La gamme est ensuite définie avec l'aide des attributs "*minimum*", "*maximum*" et "*granularity*". Ce dernier n'est pas nécessaire pour les attributs contenant des nombres flottants.

Pour chaque instance de *ManagedElement* représentant un élément de réseau, on peut créer une ou plusieurs instances de *AttributeRanges*. Celles-ci sont liées à l'instance de *ManagedElement* au moyen d'une relation de confinement.

Les gammes sont définies par leur instance de *ManagedElement*. Cela permet à un attribut de disposer de plusieurs gammes selon qu'ils se rapportent à des éléments de réseau différents. Autrement dit, le champ de chaque instance de *AttributeRanges* est composé des objets pertinents associés à l'instance de *ManagedElement* contenant l'instance de *AttributeRanges*.

La Figure 1 illustre clairement ce concept de limitation des champs. Dans la figure, nous observons deux instances différentes de *ManagedElement* (A et B). On trouve dans *ManagedElement* A deux instances de *AttributeRanges* désignées A et B. D'une manière analogue, *ManagedElement* B contient deux autres instances de *AttributeRanges* désignées C et D. *AttributeRanges* A définit les champs de toutes les instances de *AalProfileTypeOne* associées à *ManagedElement* A, alors que *AttributeRanges* C définit les champs de toutes les instances de *AalProfileTypeOne* associées à *ManagedElement* B. D'une manière analogue, *AttributeRanges* B définit les champs de toutes les

instances de *CesServiceProfile* associées à *ManagedElementA* et *AttributeRanges D* définit les champs des instances de *CesServiceProfile* associées à *ManagedElement B*. Autrement dit, le système de gestion instancie une instance de *AttributeRanges* (présentée sous forme de tableau dans la figure) par classe et par instances d'élément géré.

Dès lors, si le système de gestion doit modifier les paramètres d'une instance de *AalProfileTypeOne* associée à *ManagedElement A* (par exemple l'instance de *ProfileA* dans la figure), il peut questionner *AttributeRanges A* avant de modifier les valeurs.

Pour établir les champs des attributs définis dans les structures de données, on utilise la notation "par point". Considérons, par exemple, la structure de données suivante:

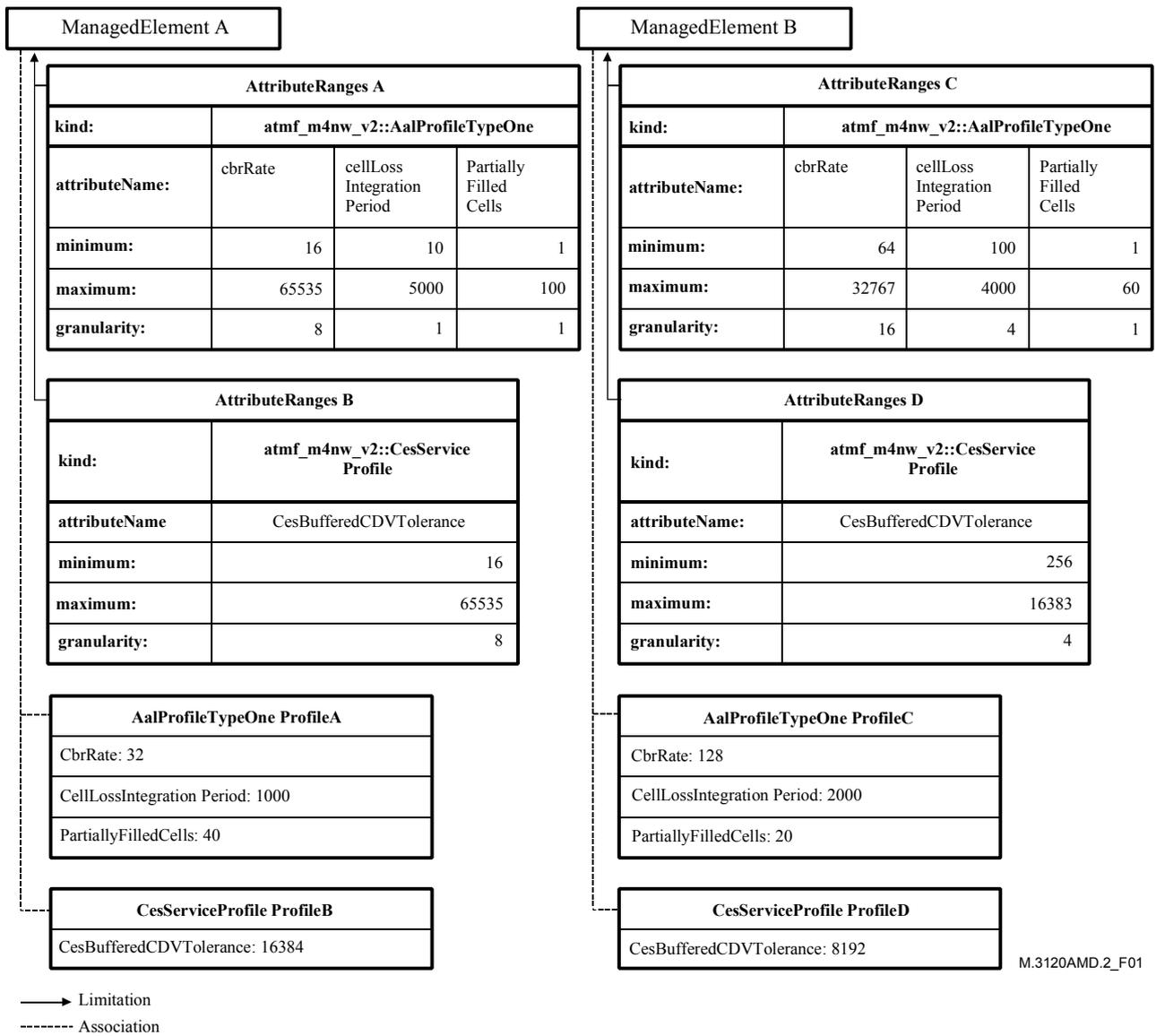
```
struct SampleStructureType {  
    long xyz,  
    long abc,  
    float def };
```

Pour définir un champ d'attribut pour l'attribut *xyz*, on peut se référer à l'attribut *xyz* en définissant l'attribut *attributeName* de *AttributeRangeType* comme étant "*SampleStructureType.xyz*".

Le § 10.2.1 définit un ensemble d'interfaces IDL CORBA pour le modèle d'information des champs de valeurs d'attribut. Ces interfaces sont traduites manuellement à partir d'un ensemble de classes d'objets gérés GDMO conformes à l'Amendement 7/M.3100 en respectant le cadre CORBA RGT et les directives données dans les Recommandations UIT-T Q.816 et X.780 pour les interfaces CORBA à granulation fine.

Outre l'interface à granulation fine du § 10.2.1, une interface de façade complémentaire est définie au § 10.3.1. Cette interface de façade est définie en fonction du cadre à granulation fine et des directives données dans les Recommandations UIT-T Q.816.1 et X.780.1 pour la prise en charge des interfaces CORBA à granulation grossière. Le nom de cette interface de façade est celui de l'interface à granulation fine correspondante complété de "_F" (soulignement suivi d'un "F" majuscule).

Les Figures 2 et 3 montrent les relations d'héritage et de confinement des interfaces CORBA définies dans le présent paragraphe. On notera que les interfaces de façade ont la même relation de hiérarchie d'héritage que les interfaces à granulation fine correspondantes.



M.3120AMD.2_F01

Figure 1/M.3120/Amd.2 – Diagramme d'instance présentant l'utilisation de AttributeRanges

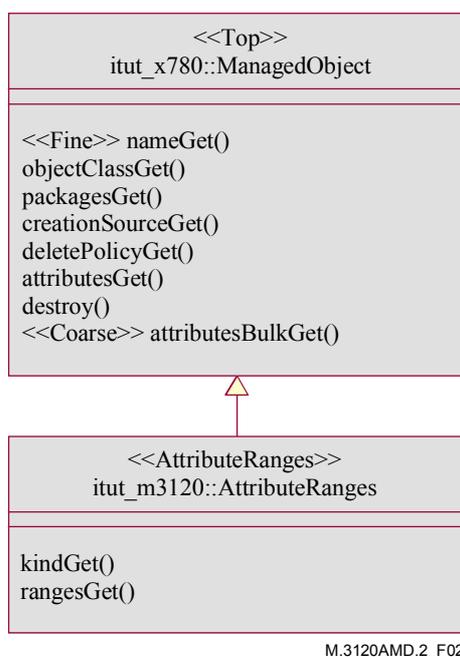


Figure 2/M.3120/Amd.2 – Relation d'héritage des gammes de valeurs d'attribut

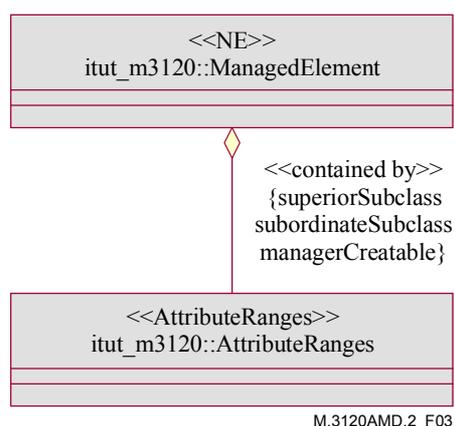


Figure 3/M.3120/Amd.2 – Relation de confinement des gammes de valeurs d'attribut

7 Aperçu général de la classe d'objets TTP de transport générique

Le présent paragraphe définit une nouvelle classe d'objets des points TTP de transport générique. Ce nouvel objet est utilisé pour représenter un port physique ou des points d'extrémité de connexions d'acheminement. Il peut être utilisé dans des modèles spécifiques à la technologie en tant qu'abstraction d'une couche de transport sous-jacente.

Une nouvelle interface GenericTransportTTP a été définie. Cet objet est une sous-classe de NetworkTP. Elle se rapporte à ManagedElement au moyen des relations de confinement. Elle est associée à CircuitPack au moyen de l'attribut PortAssociationList et à LinkEnd au moyen de l'attribut ClientLinkEndPointList.

Le § 10.2.2 définit un ensemble d'interfaces IDL CORBA pour les classes d'objets GenericTransportTTP et GenericTransportPmCD. Ces interfaces sont traduites manuellement à partir d'un ensemble de classes d'objets gérés GDMO conformément à l'Amendement 8/M.3100 compte tenu du cadre CORBA RGT et des directives données dans les Recommandations UIT-T Q.816 et X.780 pour les interfaces CORBA à granulation fine.

Outre l'interface à granulation fine du § 10.2.2, un ensemble d'interfaces de façade complémentaire est défini au § 10.3.2. Ces interfaces de façade sont définies en fonction du cadre à granulation grossière et des directives données dans les Recommandations UIT-T Q.816.1 et X.780.1 pour les interfaces supports CORBA à granulation grossière. Le nom de ces interfaces de façade est celui de l'interface à granulation fine correspondante complété de "_F" (soulignement suivi d'un "F" majuscule).

Les Figures 4 et 5 montrent les relations d'héritage, de confinement et d'association des interfaces CORBA définies dans le présent paragraphe. On notera que les interfaces de façade suivent la même relation de hiérarchie d'héritage que les interfaces à granulation fine correspondantes.

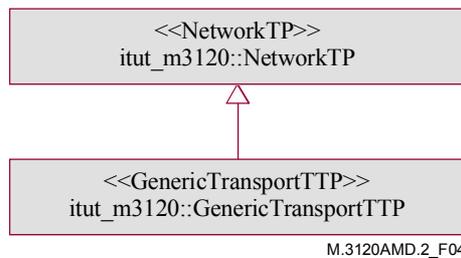


Figure 4/M.3120/Amd.2 – Relation d'héritage TTP de transport générique

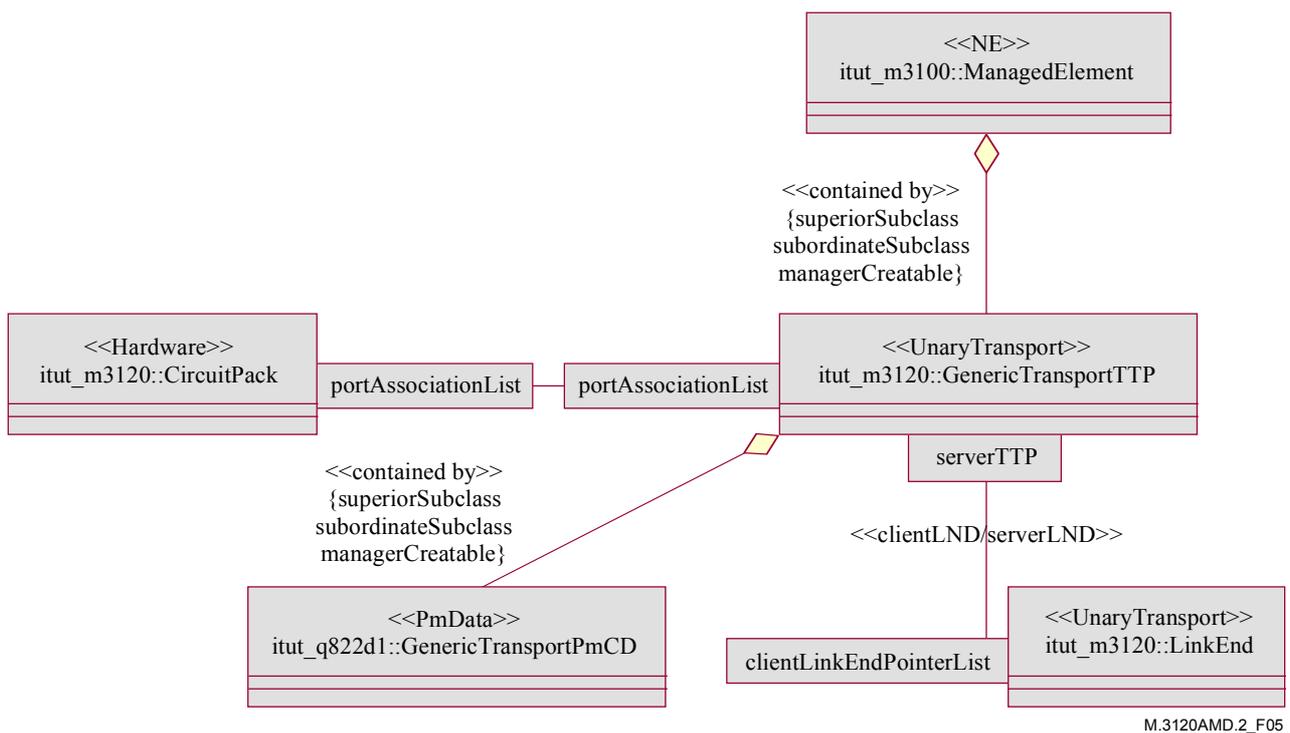


Figure 5/M.3120/Amd.2 – Relation de confinement et d'association de TTP de transport générique

8 Amélioration de la classe d'objets ManagedElement

Le présent paragraphe présente des nouveaux attributs qu'il convient d'ajouter à la classe ManagedElement. Afin de préserver la compatibilité amont, ces nouveaux attributs sont placés dans une sous-classe de ManagedElement appelée ManagedElementR2. La classe ManagedElement initiale définie dans la Rec. UIT-T M.3120 est basée sur l'objet ManagedElementR1 M.3100.

L'objet ManagedElementR2 défini ici est basé sur l'objet ManagedElementR2 de la Rec. UIT-T M.3100 défini dans l'Amendement 6/M.3100. Le nom de l'interface ManagedElementR1 est omis dans le présent amendement afin de conserver l'alignement des noms des objets M.3100 et M.3120.

L'objet ManagedElementR2 hérite de tous les attributs de ManagedElement et définit les trois éléments supplémentaires suivants:

8.1 Code du modèle

Cet attribut contient le code du modèle du produit de l'élément de réseau. Le code de modèle de produit est l'information d'identification du modèle par le fabricant. Il s'agit d'une information du fournisseur que celui-ci utilise pour distinguer l'élément de réseau dans l'ensemble de la famille des produits. Cet attribut est utile pour les systèmes OSS lorsque ceux-ci procèdent à la découverte et à l'inventaire des équipements.

Le code du modèle est un attribut en lecture seulement.

8.2 Pseudonymes des éléments de réseau

Cet attribut est utilisé pour contenir les pseudonymes attribués par l'EMS à une instance de Managed Element. Le fait de disposer de tels pseudonymes liés à l'interface EMS/NMS est utile pour mettre les noms d'éléments de réseau entrés au niveau de l'EMS, via l'interface d'utilisateur graphique ou par une autre voie en rapport avec ceux trouvés sur l'interface d'utilisateur NMS. Fonction plus importante, ces pseudonymes peuvent apparaître dans les alarmes envoyées par certains logiciels EMS à l'extérieur de l'interface. Il est donc très important que le NMS reconnaisse de tels pseudonymes afin de pouvoir exécuter les fonctions d'alarme et autres se rapportant aux défaillances et aux performances.

8.3 Type d'élément de réseau

Actuellement, Managed Element ne contient aucun attribut pour spécifier le type d'élément de réseau qu'il représente. Cet attribut contient un ensemble de chaînes textuelles ou de valeurs faisant partie d'un ensemble prédéfini (UID) qui décrit le type générique de l'élément Network Element modélisé par l'instance de ManagedElementR2. On peut utiliser plusieurs valeurs de type d'élément géré pour décrire un équipement hybride. Le type d'élément de réseau est un attribut en lecture seulement.

9 Développement des informations sur les caractéristiques

En sa forme actuelle, le module des constantes d'information des caractéristiques ne tient pas compte d'un grand nombre de débits de signalisation couramment utilisés. On trouvera ci-après un développement de la liste des débits de signal afin qu'il soit possible de décrire de manière appropriée tous les débits de signal et types de ports possibles.

Il est nécessaire de faire les adjonctions suivantes à la Rec. UIT-T M.3120 pour développer la liste du type Characteristic Info:

Paragraphe 7.2

Ajouter les lignes suivantes au module CharacteristicInfoConst:

```
const short E5_565M = 24;
const short STS3c_and_VC4_1c = 25;
const short STS12c_and_VC4_4c = 26;
const short STS48c_and_VC4_16c = 27;
const short STS192c_and_VC4_64c = 28;
const short Section_OC1_STS1_and_RS_STM0 = 29;
const short Section_OC192_STS192_and_RS_STM64 = 30;
```

```

const short Line_OC1_STS1_and_MS_STM0           = 31;
const short Line_OC192_STS192_and_MS_STM64      = 32;
const short FC_12_133M                          = 33;
// Fiber Channel protocol,
const short FC_25_266M                          = 34;
// Fiber Channel protocol,
const short FC_50_531M                          = 35;
// Fiber Channel protocol,
const short FC_100_1063M                       = 36;
// Fiber Channel protocol,
const short FDDI                                = 37;
const short Fast_Ethernet                      = 38;
const short Gigabit_Ethernet                   = 39;
const short ISDN_BRI                           = 40;
// ISDN Basic Rate Interface PTP layer rate
const short DSR_OC192_and_STM64                = 41;
const short DSR_OC768_and_STM256               = 42;
const short Section_OC24_STS24_and_RS_STM8     = 43;
const short Line_OC24_STS24_and_MS_STM8        = 44;
const short Section_OC768_STS768_and_RS_STM256 = 45;
const short Line_OC768_STS768_and_MS_STM256    = 46;
const short 10Gigabit_Ethernet                 = 47;

```

10 Modèle d'information

Le langage IDL du présent amendement fait partie intégrante de la Rec. UIT-T M.3120. Cela revient à dire que toutes les définitions (classes d'objets, type, structure) de la Rec. UIT-T M.3120 se trouvent dans le même module IDL et que l'on peut s'y référer sans identificateur de module.

La compilation du langage IDL du présent amendement a été menée à bonne fin sans erreur de syntaxe. Le compilateur utilisé est réputé conforme CORBA 2.3, qui englobe le type de valeur et les capacités macro M4.

```

#ifndef _itut_m3120_amd2_idl_
#define _itut_m3120_amd2_idl_

```

```

#include <itut_m3120.idl>

```

```

#pragma prefix "itu.int"

```

```

/**

```

```

Ce code IDL (commençant par la ligne "#ifndef ... " jusqu'à la fin du présent
paragraphe) est destiné à être enregistré dans un fichier appelé
"itut_m3120_amd2.idl" situé sur le trajet de recherche utilisé par le
compilateur IDL sur votre système. Il faut utiliser un compilateur qui supporte
la version CORBA version spécifiée dans la Rec. UIT-T Q.816. Le module principal
M.3120 (spécifié dans la Rec. UIT-T M.3120) est contenu dans un fichier distinct
appelé "itut_m3120.idl"
*/

```

```

*/

```

```

/**

```

```

Ce fragment est ajouté au module, itut_m3120, qui contient la définition IDL
basée sur des objets définis dans la Rec. UIT-T M.3100.
*/

```

```

*/

```

```
module itut_m3120
{
```

```
/**
```

10.1 Structures et TypeDefs

```
*/
```

```
enum AttributeChoiceType
{
    attributeChoiceLong,
    attributeChoiceLongLong,
    attributeChoiceUnsignedLong,
    attributeChoiceShort,
    attributeChoiceFloat
};
```

```
union AttributeRangeType switch (AttributeChoiceType)
{
    case attributeChoiceLong:
        Istring    attributeName;
        long       minimumValue;
        long       maximumValue;
        long       granularity;
        // 0 indique que cet attribut n'est pas utilisé

    case attributeChoiceLongLong:
        Istring    attributeName;
        long long   minimumValue;
        long long   maximumValue;
        long long   granularity;
        // 0 indique que cet attribut n'est pas utilisé

    case attributeChoiceUnsignedLong:
        Istring    attributeName;
        unsigned long   minimumValue;
        unsigned long   maximumValue;
        unsigned long   granularity;
        // 0 indique que cet attribut n'est pas utilisé

    case attributeChoiceShort:
        Istring    attributeName;
        short      minimumValue;
        short      maximumValue;
        short      granularity;
        // 0 indique que cet attribut n'est pas utilisé

    case attributeChoiceFloat:
        Istring    attributeName;
        float      minimumValue;
        float      maximumValue;
};
```

```
typedef sequence<AttributeRangeType> AttributeRangeSetType;
```

```
enum MeTypeChoiceType
{
    MeTypeChoiceIstring,
    MeTypeChoiceUID
};
```

```

union MeType switch (MeTypeChoiceType)
{
    case MeTypeChoiceIstring:
        Istring    MeTypeString;

        // 0 indique que cet attribut n'est pas utilisé
        DefaultLongTypeOpt defaultValue;

    case MeTypeChoiceUID:
        UIDType    METypeUID;

};

typedef sequence<MeType> MeTypeSetType;

/**
La structure de l'ID du port, managedElement et le port sont nécessaires, les
autres éléments sont facultatifs
*/
struct PortIDType
{
    Istring managedElement;
    Istring bay;
    Istring shelf;
    Istring drawer;
    Istring slot;
    Istring port;
};

/**
Déclarations de Interface vers l'avant
*/
interface AttributeRanges;
interface GenericTransportTTP;
interface ManagedElementR2;

/**
Déclarations Valuetype vers l'avant
*/
valuetype AttributeRangesValueType;
valuetype GenericTransportTTPValueType;
valuetype ManagedElementR2ValueType;

/**
Déclarations Typedefs vers l'avant
*/
typedef MOnNameType AttributeRangesNameType;
typedef MOnNameType GenericTransportTTPNameType;
typedef MOnNameType ManagedElementR2NameType;

/**
Exceptions pour paquetages conditionnels
*/
exception NONeAliasPackage {};
exception NOPortIdPackage {};

/**

```

10.2 Interfaces – A granulation fine

*/

/**

10.2.1 AttributeRanges

La classe AttributeRanges permet au système de gestion de signaler les valeurs minimales et maximales qu'accepte un attribut donné ainsi que la granularité, ou valeur de progression, du champ. Chaque instance de AttributeRanges contient des champs pour les attributs appartenant à une classe d'objets. L'attribut "kind" de AttributeRanges désigne la classe d'objets pour laquelle les champs sont définis. "attributeName" spécifie le nom de l'attribut pour lequel un champ est défini. Le champ est alors défini au moyen des attributs "minimum", "maximum" et "granularity" ("granularity" n'est pas prévu pour les types "float" car ils ne sont pas nécessaires).

Pour chaque instance de ManagedElement représentant un élément de réseau, on peut créer une ou plusieurs instances AttributeRanges. Celles-ci sont liées à l'instance ManagedElement via une relation de confinement.

Les champs sont définis par instance de ManagedElement. Cela permet à un attribut d'avoir des champs différents selon qu'il se rapporte à des éléments de réseau différents. Autrement dit, le champ de chaque instance de AttributeRanges est l'objet pertinent associé à ManagedElement qui contient l'instance AttributeRanges.

```
        valuetype AttributeRangesValueType: truncatable
itut_x780::ManagedObjectValueType
{
    public Istring          kind;
        // GET
    public attributeRangeSetType  ranges;
        // GET
}; // valuetype AttributeRangesValueType
```

```
interface AttributeRanges: itut_x780::ManagedObject
{
```

/**

Cette opération sert à obtenir la classe d'objets pour laquelle l'instance de AttributeRanges définit les champs d'attribut. La valeur renvoyée est une chaîne contenant un nom de classe d'objets.

*/

```
    Istring kindGet()
        raises (itut_x780::ApplicationError);
```

/**

Cette opération sert à obtenir l'ensemble de champs d'attributs pour la classe définie dans l'attribut "kind". La valeur renvoyée est une structure de AttributeRangesType contenant chacune le nom d'attribut, les bonds maxima et minima ainsi que la granulométrie, ou valeurs de progression, possible dans les bonds.

*/

```
    AttributeRangesSetType rangesGet()
        raises (itut_x780::ApplicationError);
```

```
MANDATORY_NOTIFICATION(
    itut_x780::Notifications, objectCreation)
```

```

MANDATORY_NOTIFICATION(
    itut_x780::Notifications, objectDeletion)
MANDATORY_NOTIFICATION(
    itut_x780::Notifications, attributeValueChange)

}; // interface AttributeRanges

```

```

/**
Fabrique de AttributeRanges
Il est prévu que cet objet soit créé au stade de l'initialisation par le système
géré.
*/

```

```

interface AttributeRangesFactory: itut_x780::ManagedObjectFactory
{
    AttributeRanges create
        (in NameBindingType nameBinding,
         in MOnNameType superior,
         inout Istring name, // auto naming if empty string
         in StringSetType packageNameList)
        raises (itut_x780::ApplicationError,
               itut_x780::CreateError);
}; // interface AttributeRangesFactory

```

```

/**

```

10.2.2 Point TTP de transport générique

```

*/
/**
Cet objet sert à représenter la liste des ports physiques.
*/

```

```

valuetype GenericTransportTTPValueType: truncatable
itut_m3120::NetworkTPValueType
{
/** GenericTransportTTPValueType utilise la forme héritée suivante
itut_m3120:: TPValueType:
    public MOnNameSetType          supportedByObjectList;
        // pointe vers le paquet de circuits prenant en charge
        // GET
    public OperationalStateType    operationalState;
        // conditionnel, présent si une instance le prend en charge.
        // GET
    public AlarmStatusType         alarmStatus;
// conditionnel, présent si le point TP prend en charge la notification
// d'alarme des communications.
    // GET
    public CurrentProblemSetType    currentProblemList;
        // conditionnel, présent si le point TP prend en charge la
        // notification d'alarme des communications.
        // GET
    public AlarmSeverityAssignmentProfileNameType
        alarmSeverityAssignmentProfilePointer;
        // conditionnel, présent si une instance prend en charge
        // la configuration de la gravité des alarmes.
        // GET-REPLACE

```

```

GenericTransportTTPValueType uses the following inherited from
itut_m3120:: NetworkTPValueType:
    public PointDirectionalityType pointDirectionality;

```

```

    // GET
    public SignalIdType      signalId;
    // GET, SET-BY-CREATE

*/

    public PortIDType PortID;
    // conditional
    // PortIdPackage
    // présent si le port TTP serveur est représenté
    // GET

    public MOnameSetType clientLinkEndPointerList;
    // GET-REPLACE

    public PointCapacityType potentialCapacity;
    // conditional
    // présent si le TTP est une technologie à débit adaptatif
    // GET

}; // valuetype GenericTransportTTPValueType

/**
Définition d'interface à granulation fine
*/

interface GenericTransportTTP: itut_m3120::NetworkTP
{

    PortIDType portIDGet ()
        raises (itut_x780::ApplicationError,
            NOPortIdPackage);

    MOnameSetType clientLinkEndPointerListGet ()
        raises (itut_x780::ApplicationError);
    void clientLinkEndPointerListSet
        (in MOnameSetType clientLinkEndPointerList)
        raises (itut_x780::ApplicationError);

    PointCapacityType potentialLinkEndCapacityPackageGet ()
        raises (itut_x780::ApplicationError);

    MANDATORY_NOTIFICATION (
        itut_x780::Notifications, objectCreation)
    MANDATORY_NOTIFICATION (
        itut_x780::Notifications, objectDeletion)
    MANDATORY_NOTIFICATION (
        itut_x780::Notifications, attributeValueChange)
    MANDATORY_NOTIFICATION (
        itut_x780::Notifications, stateChange)
    MANDATORY_NOTIFICATION (
        itut_x780::Notifications, communicationAlarm)

}; // interface GenericTransportTTP

```

```

interface GenericTransportTTPFactory:
  itut_x780::ManagedObjectFactory
{
  itut_x780::ManagedObject create
    (in NameBindingType nameBinding,
     // nom du module contenant l'information de corrélation de noms.
     in MOnameType superior,
     // Nom de l'objet contenant.
     in string reqID,
     // Valeur ID demandée pour le nom; sera vide
     // s'il faut utiliser l'attribution automatique de noms.
     out MOnameType name,
     // Nom complet d'objet nouvellement créé.
     in StringSetType packageNameList,
     // Liste des paquetages requis.

     in MOnameSetType supportedByObjectList,
     // éventuellement zéro
     in AlarmSeverityAssignmentProfileNameType
     alarmSeverityAssignmentProfilePointer,
in PointDirectionalityType pointDirectionality,

     in MOnameSetType clientLinkEndPointList)

    raises (itut_x780::ApplicationError,
           itut_x780::CreateError);

}; // interface GenericTransportTTPFactory

```

/**

10.2.3 ManagedElementR2

Les objets ManagedElementR2 sont des objets gérés qui représentent des équipements de télécommunication ou des entités RGT (groupes ou parties) dans les réseaux de télécommunication qui assurent des fonctions d'élément géré, autrement dit qui assurent de l'aide ou un service à l'abonné. Les objets gérés peuvent ou ne peuvent pas en outre assurer des fonctions de médiation/OS. Un élément géré communique avec le gestionnaire via des interfaces CORBA normalisées pour les besoins de surveillance et/ou de contrôle. Un élément géré contient des équipements qui peuvent ou ne peuvent pas être géographiquement répartis.

Lorsque l'objet élément géré assure des notifications de changement de valeur d'attribut, la notification de attributeValueChange sera émise quand la valeur de l'un des attributs suivants change: situation d'alarme, étiquette d'utilisateur, version, nom d'emplacement, liste des problèmes en cours et de l'alarme locale visuelle/sonore enclenchement. En ce qui concerne les attributs ci-dessus qui ne sont pas pris en charge, l'attitude relative à l'émission de notification de changement de valeur d'attribut s'applique uniquement lorsque l'attribut est assuré par l'objet géré. Lorsque l'objet prend en charge les notifications de changement d'état, la stateChangeNotification sera émise si la valeur de l'état administratif, opérationnel ou d'utilisation change.

La suppression par protocole de gestion n'est pas permise (l'objet devrait lancer une exception DeleteNotAllowed exception en réponse à une opération de suppression).

Cette interface est basée sur l'objet Managed Element R2 de l'Amendment 6/M.3100. Le nom de l'interface ManagedElementR1 est omis dans le présent amendement par souci d'alignement des noms de objets M.3100 et M.3120.

Ce valuetype est utilisé pour rechercher tous les attributs ManagedElementR2 en une seule opération. La plupart des attributs non pris en charge seront renvoyés comme une chaîne ou liste. La réception d'une chaîne vide ne signifie toutefois pas que l'attribut n'est pas pris en charge.

```
*/  
  
    valuetype ManagedElementR2ValueType: truncatable  
itut_m3120::ManagedElementValueType  
{  
    public Istring          modelCode;  
        // GET  
    public IstringSetType   neAliases;  
        // conditional  
        // neAliasPackage  
        // GET  
    public MeTypeSetType    managedElementType;  
        // GET  
  
}; // valuetype ManagedElementR2ValueType  
  
interface ManagedElementR2: itut_m3120::ManagedElement  
{  
/**  
La méthode suivante est employée pour l'envoi du code du modèle de produit de  
l'élément de réseau. Le code de modèle de produit est l'information  
d'identification du modèle par le fabricant. Il s'agit d'une information du  
fournisseur que celui-ci utilise pour distinguer l'élément de réseau dans  
l'ensemble de la famille des produits. Cet attribut est utile pour les systèmes  
OSS lorsque ceux-ci procèdent à la découverte et à l'inventaire des équipements.  
*/  
    Istring modelCodeGet ()  
        raises (itut_x780::ApplicationError);  
/**  
La méthode suivante renvoie un ensemble de chaînes contenant des pseudonymes de  
l'élément géré tel qu'il est défini par l'EMS. Ces pseudonymes sont donnés par  
l'EMS à une instance d'élément géré donnée. Le fait de disposer de tels  
pseudonymes liés à l'interface EMS/NMS est utile pour mettre les noms d'éléments  
de réseau entrés au niveau de l'EMS, via l'interface d'utilisateur graphique ou  
par une autre voie en rapport avec ceux trouvés sur l'interface  
d'utilisateur NMS. Fonction plus importante, ces pseudonymes peuvent apparaître  
dans les alarmes envoyées par certains logiciels EMS à l'extérieur de  
l'interface.  
*/  
    Istring neAliasesGet ()  
        raises (itut_x780::ApplicationError,  
                NOneAliasPackage);  
/**  
Le procédé suivant renvoie un ensemble de chaînes de texte et d'UID décrivant le  
type générique de l'élément de réseau modélisé par l'instance ManagedElementR2.  
Plusieurs valeurs de type d'élément géré peuvent être utilisées pour décrire un  
équipement hybride.  
*/  
    MeTypeSetType managedElementTypeGet ()  
        raises (itut_x780::ApplicationError);  
  
    CONDITIONAL_NOTIFICATION(  
        itut_x780::Notifications, objectCreation,  
        createDeleteNotificationsPackage)  
    CONDITIONAL_NOTIFICATION(  
        itut_x780::Notifications, objectDeletion,  
        createDeleteNotificationsPackage)
```

```

CONDITIONAL_NOTIFICATION(
    itut_x780::Notifications, attributeValueChange,
    attributeValueChangeNotificationPackage)
CONDITIONAL_NOTIFICATION(
    itut_x780::Notifications, stateChange,
    stateChangeNotificationPackage)
CONDITIONAL_NOTIFICATION(
    itut_x780::Notifications, stateChange,
    stateChangeNotificationPackage)

}; // interface ManagedElementR2

interface ManagedElementR2Factory: itut_x780::ManagedObjectFactory
{
    ManagedElementR2 create
        (in NameBindingType nameBinding,
         in MOnameType superior,
         inout Istring name, // autodénomination si la chaîne est vide
         in StringSetType packageNameList,
         in AdministrativeStateType administrativeState,
          // managedElementPackage
          // GET-REPLACE
         in boolean enableAudibleVisualLocalAlarm,
          // conditional
          // audibleVisualLocalAlarmPackage
          // GET-REPLACE
         in AlarmSeverityAssignmentProfileNameType profile,
          // conditional
          // alarmSeverityAssignmentPointerPackage
          // GET-REPLACE
         in Istring userLabel,
          // conditional
          // userLabelPackage
          // GET-REPLACE
         in Istring vendorName,
          // conditional
          // vendorNamePackage
          // GET-REPLACE
         in Istring version,
          // conditional
          // versionPackage
          // GET-REPLACE
         in Istring locationName,
          // conditional
          // locationNamePackage
          // GET-REPLACE
         in ExternalTimeType externalTime,
          // conditional
          // externalTimePackage
          // GET-REPLACE
         in SystemTimingSourceType systemTimingSource,
          // conditional
          // systemTimingSourcePackage
          // GET-REPLACE
         in ArcProbableCauseSetType arcProbableCauseList,
          // conditional
          // arcPackage
          // GET-REPLACE, ADD-REMOVE
         in ArcIntervalProfileNameType arcIntervalProfilePointer,
          // conditional
          // arcPackage
          // GET-REPLACE

```

```

        in ArcTimeType arcManagementRequestedInterval)
        // conditional
        // arcPackage
        // GET-REPLACE
        raises (itut_x780::ApplicationError,
            itut_x780::CreateError);

}; // interface ManagedElementR2Factory

```

```
/**
```

10.3 Interfaces – Façade

Le comportement des interfaces de façade est identique à celui des interfaces à granulation fine correspondantes, raison pour laquelle ces interfaces de façade ne comportent pas de commentaires. Les lecteurs se référeront à l'interface à granulation fine du § 10.2 au sujet du comportement de ces interfaces de façade.

Le présent paragraphe peut être omis de l'IDL si le système de gestion n'accepte que les interfaces à granulation fine.

```
*/
```

```
/**
```

10.3.1 AttributeRanges_F

```
*/
```

```

interface AttributeRanges_F: itut_x780::ManagedObject_F
{
    Istring kindGet(in MOnameType name)
        raises (itut_x780::ApplicationError);

    attributeRangesSetType rangesGet(in MOnameType name)
        raises (itut_x780::ApplicationError);

    MANDATORY_NOTIFICATION(
        itut_x780::Notifications, objectCreation)
    MANDATORY_NOTIFICATION(
        itut_x780::Notifications, objectDeletion)
    MANDATORY_NOTIFICATION(
        itut_x780::Notifications, attributeValueChange)
}; // interface AttributeRanges_F

```

```
/**
```

10.3.2 GenericTransportTTP_F

```
*/
```

```
/**
```

```
Coarse-Grained Interface Definition
```

```
*/
```

```

interface GenericTransportTTP_F: itut_m3120::NetworkTTP_F
{

```

```

/**
Des instances de GenericTransportTTP sont créées au moyen de
GenericTransportTTPFactory ou automatiquement par le système géré.
*/

/** GenericTransportTTP_F hérite des méthodes suivantes de
itut_m3120::TP_F:
supportedByObjectListGet,
operationalStateGet, alarmStatusGet, containedInSubnetworkListGet,
currentProblemListGet, alarmSeverityAssignmentProfilePointerGet,
alarmSeverityAssignmentProfilePointerSet

GenericTransportTTP_F hérite des méthodes suivantes de
itut_m3120:: NetworkTP_F:
pointDirectionalityGet, signalIdGet
*/

PortIDType portIDGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
           NOPortIdPackage);

MOnameSetType clientLinkEndPointListGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError);
void clientLinkEndPointListSet
    (in MOnameType name,
     in MOnameSetType clientLinkEndPointList)
    raises (itut_x780::ApplicationError);

/**

Offre la largeur de bande potentielle pour la technique de serveur à débit
adaptatif.
*/
PointCapacityType potentialLinkEndCapacityPackageGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError);

MANDATORY_NOTIFICATION(
    itut_x780::Notifications, objectCreation)
MANDATORY_NOTIFICATION(
    itut_x780::Notifications, objectDeletion)
MANDATORY_NOTIFICATION(
    itut_x780::Notifications, attributeValueChange)
MANDATORY_NOTIFICATION(
    itut_x780::Notifications, stateChange)
MANDATORY_NOTIFICATION(
    itut_x780::Notifications, communicationAlarm)

}; // interface GenericTransportTTP_F

```

```
/**
```

10.3.3 ManagedElementR2_F

```
*/
```

```

interface ManagedElementR2_F: itut_m3120::ManagedElement_F
{

Istring modelCodeGet (in MOnameType name)
    raises (itut_x780::ApplicationError);

```

```

Istring neAliasesGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError,
           NOneAliasPackage);

Istring managedElementTypeGet
    (in MOnameType name)
    raises (itut_x780::ApplicationError);

CONDITIONAL_NOTIFICATION(
    itut_x780::Notifications, objectCreation,
    createDeleteNotificationsPackage)
CONDITIONAL_NOTIFICATION(
    itut_x780::Notifications, objectDeletion,
    createDeleteNotificationsPackage)
CONDITIONAL_NOTIFICATION(
    itut_x780::Notifications, attributeValueChange,
    attributeValueChangeNotificationPackage)
CONDITIONAL_NOTIFICATION(
    itut_x780::Notifications, stateChange,
    stateChangeNotificationPackage)

}; // interface ManagedElementR2_F

```

```
/**
```

10.4 Corrélation de noms

```
*/
```

```
/**
```

Le module suivant contient des informations de corrélation de noms.

```
*/
```

```

module NameBinding
{

```

```
/**
```

10.4.1 AttributeRanges

```
*/
```

```

module AttributeRanges_ManagedElement
{
    const string      superiorClass =
        "itut_m3120::ManagedElement";
    const boolean    superiorSubclassesAllowed = TRUE;
    const string      subordinateClass =
        "itut_m3120::AttributeRanges";
    const boolean    subordinateSubclassesAllowed = TRUE;
    const boolean    managerCreatesAllowed = FALSE;
    const DeletePolicyType deletePolicy =
        itut_x780::notDeletable;
    const string      kind = "AttributeRanges";
}; // module AttributeRanges_ManagedElement

```

```
/**
```

10.4.2 GenericTransportTTP

```
*/
```

```

module GenericTransportTTP_ManagedElement
{
    const string      superiorClass =

```

```
        "itut_m3120::ManagedElement";
const boolean superiorSubclassesAllowed = TRUE;
const string subordinateClass =
    "itut_m3120::GenericTransportTTP";
const boolean subordinateSubclassesAllowed = TRUE;
const boolean managerCreatesAllowed = FALSE;
const DeletePolicyType deletePolicy =
    itut_x780::notDeletable;
const string kind = "GenericTransportTTP";
}; // module GenericTransportTTP_ManagedElement

}; // module NameBinding

}; // module itut_m3120

#endif // _itut_m3120_amd2_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