



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

M.3120

Enmienda 2

(03/2003)

SERIE M: RGT Y MANTENIMIENTO DE REDES:
SISTEMAS DE TRANSMISIÓN, CIRCUITOS
TELEFÓNICOS, TELEGRAFÍA, FACSIMIL Y CIRCUITOS
ARRENDADOS INTERNACIONALES

Red de gestión de las telecomunicaciones

Modelo genérico de información a nivel de red y
de elemento de red basado en arquitectura de
intermediario de petición de objeto común

Enmienda 2

Recomendación UIT-T M.3120 (2001) – Enmienda 2

RECOMENDACIONES UIT-T DE LA SERIE M

RGT Y MANTENIMIENTO DE REDES: SISTEMAS DE TRANSMISIÓN, CIRCUITOS TELEFÓNICOS, TELEGRAFÍA, FACSIMIL Y CIRCUITOS ARRENDADOS INTERNACIONALES

Introducción y principios generales de mantenimiento y organización del mantenimiento	M.10–M.299
Sistemas internacionales de transmisión	M.300–M.559
Circuitos telefónicos internacionales	M.560–M.759
Sistemas de señalización por canal común	M.760–M.799
Circuitos internacionales utilizados para transmisiones de telegrafía y de telefotografía	M.800–M.899
Enlaces internacionales arrendados en grupo primario y secundario	M.900–M.999
Circuitos internacionales arrendados	M.1000–M.1099
Sistemas y servicios de telecomunicaciones móviles	M.1100–M.1199
Red telefónica pública internacional	M.1200–M.1299
Sistemas internacionales de transmisión de datos	M.1300–M.1399
Designaciones e intercambio de información	M.1400–M.1999
Red de transporte internacional	M.2000–M.2999
Red de gestión de las telecomunicaciones	M.3000–M.3599
Redes digitales de servicios integrados	M.3600–M.3999
Sistemas de señalización por canal común	M.4000–M.4999

Para más información, véase la Lista de Recomendaciones del UIT-T.

Recomendación UIT-T M.3120

Modelo genérico de información a nivel de red y de elemento de red basado en arquitectura de intermediario de petición de objeto común

Enmienda 2

Resumen

La presente enmienda aporta varias mejoras al modelo genérico de información a nivel de red y de elemento de red (NE) basado en arquitectura de intermediario de petición de objeto común. En primer lugar, se describe el mecanismo que soporta una gama de valores de atributo de generación de informes en una interfaz CORBA. En segundo lugar, se define una nueva clase de objeto de transporte genérico TTP, que pretende representar un puerto físico o puntos extremos de las conexiones de transporte. En tercer lugar, se define una nueva clase de objeto, el ManagedElementR2, que es una subclase del ManagedElement que contiene tres atributos adicionales. El primero de estos atributos contiene el código modelo (*model code*) de una parte del equipo. Otro de los nuevos atributos se utiliza para representar los alias NE, o nombres utilizados por el EMS para referirse a los elementos de red. Asimismo, se define un atributo que contiene el tipo (*type*) genérico de un NE.

Otra de las mejoras que aporta esta enmienda se refiere a la ampliación del módulo de constantes CharacteristicInfo para poder representar adecuadamente el mayor número posible de velocidades de señalización disponibles actualmente.

Orígenes

La Recomendación UIT-T M.3120 (2001), preparada por la Comisión de Estudio 4 (2001-2004) del UIT-T, fue aprobada por el procedimiento de la Resolución 1 de la AMNT el 29 de marzo de 2003.

PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Asamblea Mundial de Normalización de las Telecomunicaciones (AMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución 1 de la AMNT.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

NOTA

En esta Recomendación, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT no ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

© UIT 2003

Reservados todos los derechos. Ninguna parte de esta publicación puede reproducirse por ningún procedimiento sin previa autorización escrita por parte de la UIT.

ÍNDICE

	Página
1 Alcance	1
2 Referencias	1
3 Definiciones.....	2
4 Abreviaturas.....	2
5 Convenios	2
6 Panorámica de las gamas de valores de atributo	2
7 Panorámica del transporte genérico TTP.....	5
8 Mejoras de la clase de objeto ManagedElement.....	7
8.1 Código modelo	7
8.2 Alias de elemento de red	7
8.3 Tipo de elemento de red	7
9 Ampliación de la información característica	7
10 Modelo de información.....	8
10.1 Estructuras y TypeDefs	9
10.2 Interfaces – de granularidad fina	11
10.3 Interfaces fachada.....	17
10.4 Vinculación de nombre.....	19

Recomendación UIT-T M.3120

Modelo genérico de información a nivel de red y de elemento de red basado en arquitectura de intermediario de petición de objeto común

Enmienda 2

1 Alcance

La presente enmienda aporta varias mejoras para el modelo de información a nivel de red y de elemento de red basado en arquitectura de intermediario de petición de objeto común. En primer lugar, se describe un mecanismo que soporta una gama de valores de atributo de generación de informes en una interfaz CORBA. En segundo lugar, se define una nueva clase de objeto de transporte genérico TTP, que pretende representar un puerto físico o puntos extremos de las conexiones de transporte. En tercer lugar, se define una nueva clase de objeto, el ManagedElementR2, que es una subclase del ManagedElement que contiene tres atributos adicionales. El primero de estos atributos contiene el código modelo (*model code*) de una parte del equipo. Otro de los nuevos atributos se utiliza para representar los alias NE, o nombres utilizados por el EMS para referirse a los elementos de red. Asimismo, se define un atributo que contiene el tipo (*type*) genérico de un NE.

Otra de las mejoras que aporta esta enmienda se refiere a la ampliación del módulo de constantes CharacteristicInfo para poder representar adecuadamente el mayor número posible de velocidades de señalización disponibles actualmente.

2 Referencias

Las siguientes Recomendaciones del UIT-T y otras referencias contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y otras referencias son objeto de revisiones por lo que se preconiza que los usuarios de esta Recomendación investiguen la posibilidad de publicar las ediciones más recientes de las Recomendaciones y otras referencias citadas a continuación. Se publica periódicamente una lista de las Recomendaciones UIT-T actualmente vigentes. En esta Recomendación, la referencia a un documento, en tanto que autónomo, no le otorga el rango de una Recomendación.

- [1] Recomendación UIT-T Q. 816 (2001), *Servicios de la RGT basados en arquitectura de intermediario de petición de objeto común.*
- [2] Recomendación UIT-T Q.816.1 (2001), *Servicios de la RGT basados en arquitectura de intermediario de petición de objeto común. Extensiones para el soporte de interfaces de granularidad gruesa.*
- [3] Recomendación UIT-T X.780 (2001), *Directrices de la RGT para la definición de objetos gestionados mediante arquitectura de intermediario de petición de objeto común.*
- [4] Recomendación UIT-T X.780.1 (2001), *Directrices de la RGT para la definición de interfaces de objetos gestionados, mediante arquitectura de intermediario de petición de objeto común de granularidad gruesa.*
- [5] Recomendación UIT-T M.3120 (2001), *Modelo genérico de información a nivel de red y de elemento de red basado en arquitectura de intermediario de petición de objeto común.*

- [6] Recomendación UIT-T M.3100 (1995), *Modelo genérico de información de red*, más enmienda 1 (1999).
- [7] Recomendación UIT-T Q.822.1 (2001), *Servicio de gestión de la calidad de funcionamiento de la RGT basado en arquitectura de intermediario de petición de objeto común*.
- [8] Norma ANSI T1.231.1997 (1997), *Digital Hierarchy – Layer 1 In-Service Digital Transmission Performance Monitoring*.

3 Definiciones

En la presente enmienda no se incorporan nuevas definiciones distintas de las de la Recomendación de base.

4 Abreviaturas

En la presente enmienda no se incorporan nuevas abreviaturas distintas de las de la Recomendación de base.

5 Convenios

En la presente enmienda no se incorporan nuevos convenios distintos de las de la Recomendación de base.

6 Panorámica de las gamas de valores de atributo

En esta cláusula se describe un mecanismo que permite a los sistemas gestionados que utilizan el paradigma M.3120 generar automáticamente informes sobre las gamas de valores aceptables para los atributos asociados a un elemento de red en el modelo. Un mecanismo de este tipo puede resultar de gran valor para la detección y configuración del equipo, puesto que pone automáticamente en conocimiento del sistema de gestión las gamas de valores aceptables para cada parámetro configurable de la red antes de que se establezcan estos valores.

Para poder implementar este mecanismo, se define una nueva clase de objeto `AttributeRanges`. La clase `AttributeRanges` permite al sistema gestionado generar informes sobre los valores mínimos y máximos que aceptan determinados atributos, así como sobre la granularidad o incrementos por escalones, de la gama. Cada ejemplar `AttributeRanges` contiene gamas para los atributos que pertenecen a una clase de objeto. El atributo "*modalidad*" en el `AttributeRanges` denota la clase de objeto para la que se define cada gama. El "*nombre de atributo*" especifica el nombre del atributo para el que se define la gama. Así, la gama se define utilizando los atributos "*mínimo*", "*máximo*" y "*granularidad*". No es necesario conocer la "*granularidad*" para los atributos que contienen números en coma flotante.

Para cada ejemplar `ManagedElement` que representa un elemento de red, se pueden crear uno o más ejemplares `AttributeRanges`. Entre el ejemplar `AttributeRanges` y el ejemplar `ManagedElement` existe una relación de contención.

Las gamas se definen para cada ejemplar `ManagedElement`, lo que permite que un atributo tenga distintas gamas dependiendo del elemento de red al que pertenezca. En otras palabras, el alcance de cada ejemplar `AttributeRanges` son los objetos pertinentes asociados con el `ManagedElement` que contiene el ejemplar `AttributeRanges`.

En la figura 1 se ilustra el alcance de este concepto con más claridad. En la figura se ven dos ejemplares `ManagedElement` distintas (A y B). Dentro del `ManagedElement` A hay dos ejemplares `AttributeRanges` llamadas A y B. Del mismo modo, dentro del `ManagedElement` B hay dos ejemplares `AttributeRanges` llamadas C y D. El `AttributeRanges` A define las gamas para todos los

ejemplares `AalProfileTypeOne` asociados al `ManagedElement A`, mientras que el `AttributeRanges C` define las gamas para los ejemplares `AalProfileTypeOne` asociados al `ManagedElement B`. Del mismo modo, el `AttributeRanges B` define las gamas para todos los ejemplares `CesServiceProfile` asociadas al `ManagedElement A`, y el `AttributeRanges D` define las gamas para los ejemplares `CesServiceProfile` asociadas con el `ManagedElement B`. En otras palabras, el sistema gestionado crea un ejemplar `AttributeRanges` (ilustrado como un cuadro en la figura) por clase y por ejemplar `ManagedElement`.

Por consiguiente, si el sistema de gestión necesita modificar los parámetros de un ejemplar `AalProfileTypeOne` asociada al `ManagedElement A` (como el ejemplar *ProfileA*, en la figura), puede interrogar al `AttributeRanges A` antes de modificar los valores.

Para establecer las gamas de los atributos definidos dentro de las estructuras de datos, se utiliza la notación de punto. Por ejemplo, en la siguiente estructura de datos,

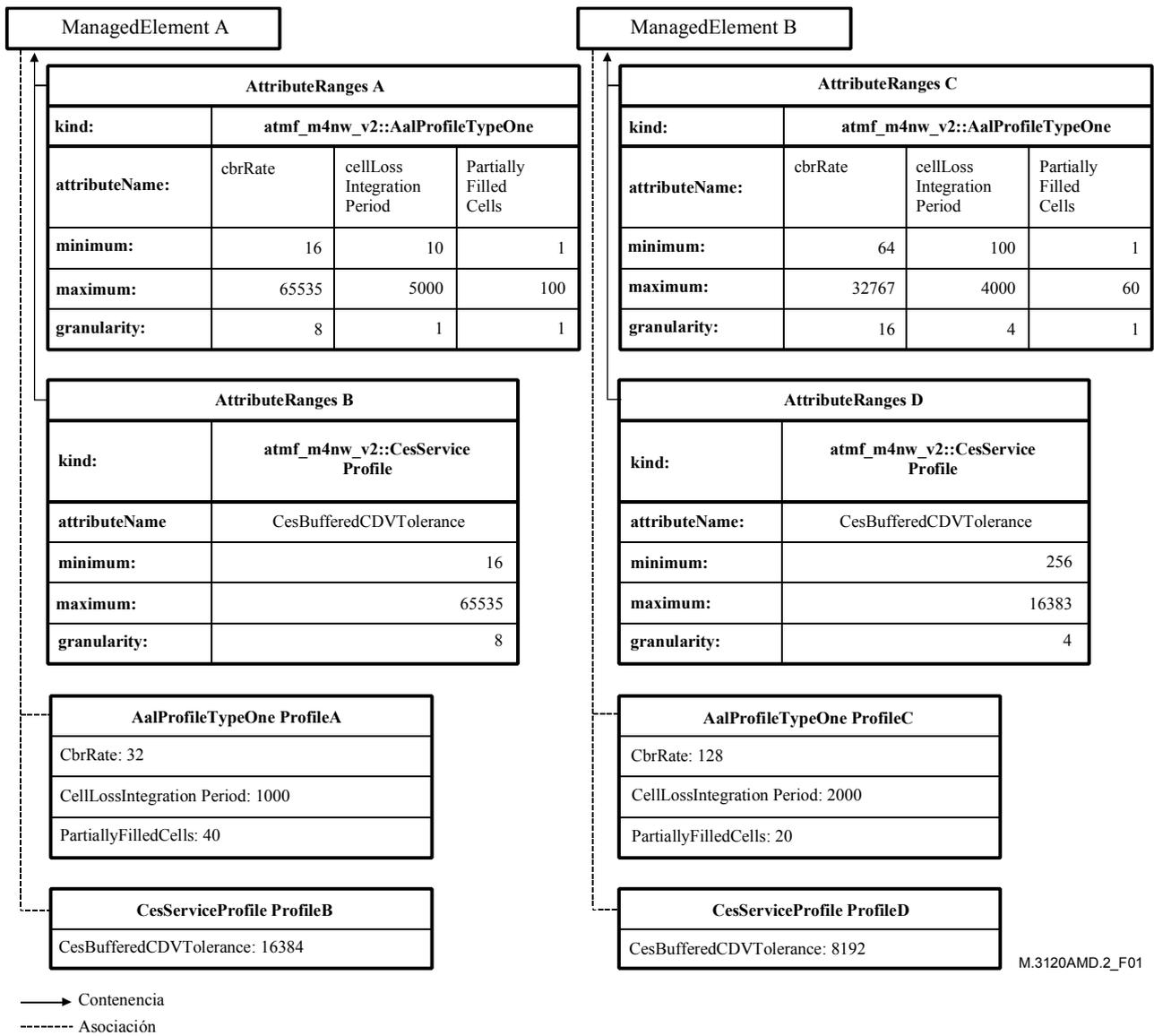
```
struct SampleStructureType {
    long xyx,
    long abc,
    float def };
```

Para establecer una gama de atributos para el atributo *xyz*, se puede hacer referencia a este último estableciendo el atributo *attributeName* dentro del *AttributeRangeType* en "*SampleStructureType.xyz*".

En la cláusula 10.2.1 se define un conjunto de interfaces CORBA IDL para el modelo de información para gamas de valores de atributo. Estas interfaces se traducen manualmente a partir de un conjunto de clases de objetos gestionados GDMO de la Enmienda 7/M.3100 siguiendo el marco y las directrices CORBA de la RGT especificados en las Recomendaciones UIT-T Q.816 y X.780 para una interfaz CORBA de granularidad fina.

Además de las interfaces de granularidad fina mencionadas en la cláusula 10.2.1, en la cláusula 10.3.1 se define un conjunto de interfaces fachada que las acompañan. Estas interfaces fachada se definen de conformidad con el marco y las directrices de granularidad gruesa dadas en las Recomendaciones UIT-T Q.816.1 y X.780.1 para el soporte de la interfaz CORBA de granularidad gruesa. Los nombres de estas interfaces fachada son los nombres de las correspondientes interfaces de granularidad fina con "_F" (un carácter de subrayado seguido de una "F" mayúscula) añadidos al final.

Las figuras 2 y 3 muestran las relaciones de herencia y contención de las interfaces CORBA definidas en esta cláusula. Cabe señalar que las interfaces fachada siguen la misma relación jerárquica de herencia que las correspondientes interfaces de granularidad fina.



M.3120AMD.2_F01

Figura 1/M.3120/Enm.2 – Ejemplar del diagrama de representación del uso del AttributeRanges

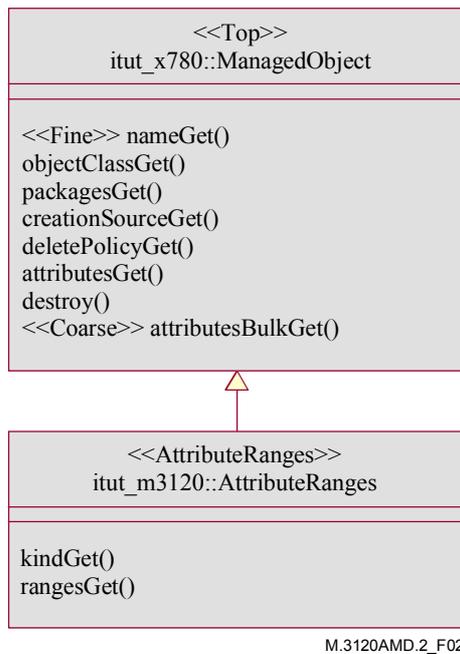


Figura 2/M.3120/Enm.2 – Relación de herencia de las gamas de valor de atributo

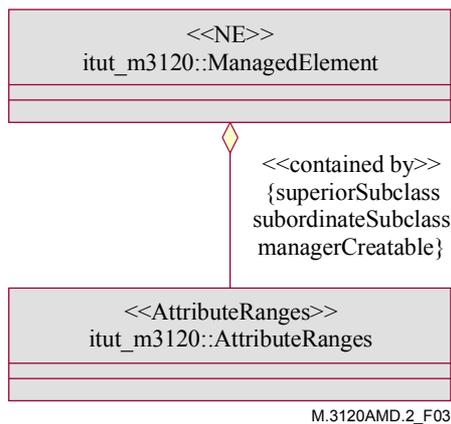


Figura 3/M.3120/Enm.2 – Relación de contención de las gamas de valor de atributo

7 Panorámica del transporte genérico TTP

En esta cláusula se define una nueva clase de objeto de transporte genérico TTP. Este nuevo objeto se utiliza para representar un puerto físico o puntos extremos de las conexiones de transporte. Se puede utilizar en modelos específicos de la tecnología como una abstracción de una capa de transporte subyacente.

Se define una nueva interfaz GenericTransportTTP. Este objeto es una subclase de la NetworkTP y mantiene con el ManagedElement una relación de contención. Está asociado con el CircuitPack mediante un atributo PortAssociationList, y con el LinkEnd mediante un atributo ClientLinkEndPointList.

En la cláusula 10.2.2 se define un conjunto de interfaces CORBA IDL para las clases de objeto GenericTransportTTP y GenericTransportPmCD. Estas interfaces se traducen manualmente a partir de un conjunto de clases de objetos gestionados GDMO de la Enmienda 8/M.3100 siguiendo el

marco y las directrices CORBA de la RGT especificados en las Recomendaciones UIT-T Q.816 y X.780 para una interfaz CORBA de granularidad fina.

Además de las interfaces de granularidad fina mencionadas en 10.2.2, en 10.3.2 se define un conjunto de interfaces fachada que las acompañan. Estas interfaces fachada se definen de conformidad con el marco y las directrices de granularidad gruesa dadas en las Recomendaciones UIT-T Q.816.1 y X.780.1 para el soporte de la interfaz CORBA de granularidad gruesa. Los nombres de estas interfaces fachada son los nombres de las correspondientes interfaces de granularidad fina con "_F" (un carácter de subrayado seguido de "F" mayúscula) añadidos al final.

Las figuras 4 y 5 muestran las relaciones de herencia, contención y asociación de las interfaces CORBA definidas en la presente Recomendación. Cabe señalar que las interfaces fachada siguen la misma relación jerárquica de herencia que las correspondientes interfaces de granularidad fina.

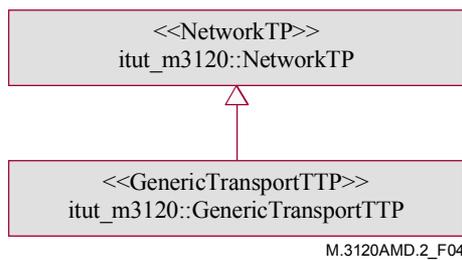


Figura 4/M.3120/Enm.2 – Relación de herencia de transporte genérico TTP

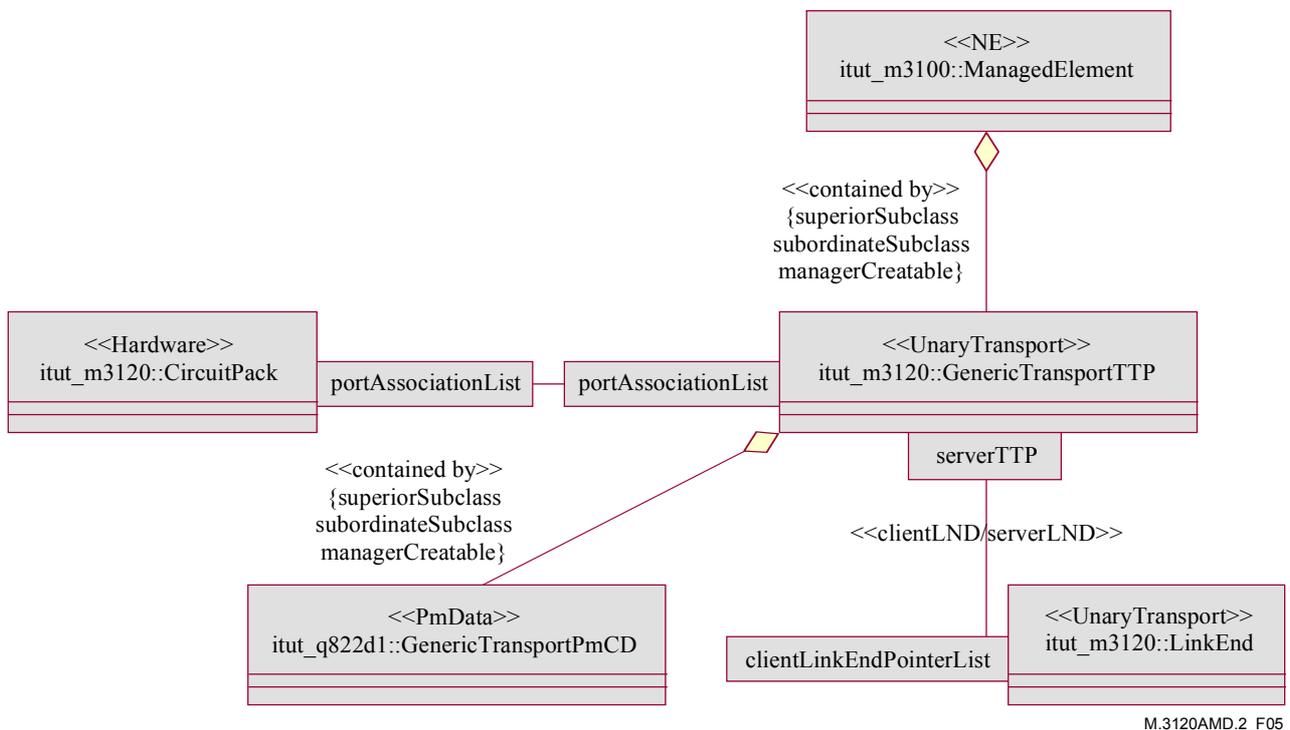


Figura 5/M.3120/Enm.2 – Relaciones de contención y asociación de transporte genérico TTP

8 Mejoras de la clase de objeto ManagedElement

En esta cláusula se describen nuevos atributos que deben añadirse a la clase ManagedElement. Para preservar la compatibilidad con versiones anteriores, estos nuevos atributos se sitúan en una subclase del ManagedElement llamada ManagedElementR2. El ManagedElement de la Rec. UIT-T M.3120 se basa en el objeto ManagedElementR1 M.3100. El objeto ManagedElementR2 que se define a continuación se basa en el objeto ManagedElementR2 de la Enmienda 6/M.3100. Se omite aquí la referencia al ManagedElementR1 para armonizar los nombres de los objetos en las Recomendaciones M.3100 y M.3120.

El ManagedElementR2 hereda todos los atributos del ManagedElement e incluye tres atributos adicionales.

8.1 Código modelo

Este atributo contiene el código modelo del producto del elemento de red. El código modelo del producto es la identificación del modelo del fabricante. Ésta es la información que presenta el proveedor y que él mismo utiliza para distinguir el elemento de red entre una familia de productos. Este atributo resulta de utilidad para los procesos de detección e inventario del equipo que lleva a cabo el OSS.

El código modelo es un atributo de sólo lectura.

8.2 Alias de elemento de red

Este atributo se utiliza para almacenar los alias que da el EMS a determinados ejemplares de elementos gestionados. Disponer de estos alias vía la interfaz EMS/NMS es útil para relacionar los nombres del elemento de red introducidos en el EMS a través de una interfaz de usuario gráfica o por cualquier otro medio, con los alias que se encuentran en la interfaz de usuario NMS. Lo que es más importante, estos alias pueden aparecer en las alarmas enviadas por determinados programas del EMS fuera de la interfaz. Así, resulta fundamental para el NMS reconocer estos alias a la hora de realizar correlaciones de alarmas u otras funciones relacionadas con las averías y la calidad de funcionamiento.

8.3 Tipo de elemento de red

En la actualidad, la clase elemento gestionado no contiene ningún atributo que especifique el tipo de elemento de red que representa. Este atributo contiene cadenas o valores textuales de un conjunto predeterminado (UID), que describe el tipo genérico del elemento de red modelado por el ejemplar ManagedElementR2. Pueden utilizarse múltiples valores de tipo de elemento gestionado para describir un equipo híbrido. El atributo de elemento de red es un atributo de sólo lectura.

9 Ampliación de la información característica

En su forma actual, el módulo de constantes de información característica excluye un gran número de velocidades de señalización ampliamente utilizadas. A continuación se amplía la lista de velocidades de señalización para que la información característica pueda describir adecuadamente el mayor número de velocidades de señalización y tipos de puerto posible.

Es necesario introducir las siguientes modificaciones en la Rec. UIT-T M.3120 para ampliar la lista de tipos de información característica:

Cláusula 7.2

Añádanse las siguientes líneas dentro del módulo 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;
    // Protocolo Fiber Channel,
const short FC_25_266M                   = 34;
    // Protocolo Fiber Channel,
const short FC_50_531M                   = 35;
    // Protocolo Fiber Channel,
const short FC_100_1063M                 = 36;
    // Protocolo Fiber Channel,
const short FDDI                         = 37;
const short Fast_Ethernet                 = 38;
const short Gigabit_Ethernet             = 39;
const short ISDN_BRI                     = 40;
    // Interfaz velocidad básica de la RDSI con velocidad de la capa PTP
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 Modelo de información

La presente enmienda del IDL forma parte integrante de la Rec. UIT-T M.3120. Esto implica que todas las definiciones (clases de objeto, tipo, estructura, etc.) de la Rec. UIT-T M.3120 son las mismas para el módulo IDL y pueden referenciarse sin necesidad de un identificador de módulo.

En esta enmienda se ha compilado satisfactoriamente el IDL sin errores de sintaxis. El compilador utilizado pretende la conformidad con CORBA 2.3, que incluye capacidades de macros de tipo de valor y M4.

```

#ifndef _itut_m3120_amd2_idl_
#define _itut_m3120_amd2_idl_

```

```

#include <itut_m3120.idl>

```

```

#pragma prefix "itu.int"

```

```

/**

```

Este código IDL (empezando con la línea "#ifndef ... ") se debe almacenar en un archivo denominado "itut_m3120_amd2.idl" ubicado en el trayecto de búsqueda utilizado por el compilador IDL de su sistema. Debe utilizarse un compilador que soporte la versión CORBA especificada en la Rec. UIT-T Q.816. El módulo principal M.3120 (definido en la Rec. UIT-T M.3120) está contenido en un archivo distinto denominado "itut_m3120.idl".

```

*/

```

```

/**

```

Se añade el siguiente fragmento al módulo itut_m3120, que contiene la definición de IDL basada en los objetos definidos en la Rec. UIT-T M.3100.

```

*/

```

```
module itut_m3120
{
```

```
/**
```

10.1 Estructuras y TypeDefs

```
*/
```

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

union AttributeRangeType switch (AttributeChoiceType)
{
    case attributeChoiceLong:
        Istring    attributeName;
        long        minimumValue;
        long        maximumValue;
        long        granularity;
        // un 0 indica que no se utiliza este atributo

    case attributeChoiceLongLong:
        Istring    attributeName;
        long long   minimumValue;
        long long   maximumValue;
        long long   granularity;
        // un 0 indica que no se utiliza este atributo

    case attributeChoiceUnsignedLong:
        Istring    attributeName;
        unsigned long    minimumValue;
        unsigned long    maximumValue;
        unsigned long    granularity;
        // un 0 indica que no se utiliza este atributo

    case attributeChoiceShort:
        Istring    attributeName;
        short      minimumValue;
        short      maximumValue;
        short      granularity;
        // un 0 indica que no se utiliza este atributo

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

typedef sequence<AttributeRangeType> AttributeRangeSetType;

enum MeTypeChoiceType
{
    MeTypeChoiceIstring,
    MeTypeChoiceUID
};
```

```

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

        // un 0 indica que no se utiliza este atributo
        DefaultLongTypeOpt defaultValue;

    case MeTypeChoiceUID:
        UIDType  MTypeUID;

};

typedef sequence<MeType> MeTypeSetType;

/**
Es necesario conocer la estructura del identificador de puerto, el
managedElement y el puerto. Los demás elementos son optativos.
*/
struct PortIDType
{
    Istring managedElement;
    Istring bay;
    Istring shelf;
    Istring drawer;
    Istring slot;
    Istring port;
};

/**
Declaraciones previas de interfaz
*/
interface AttributeRanges;
interface GenericTransportTTP;
interface ManagedElementR2;

/**
Declaraciones previas de Valuetype
*/
valuetype AttributeRangesValueType;
valuetype GenericTransportTTPValueType;
valuetype ManagedElementR2ValueType;

/**
Declaraciones previas de Typedefs
*/
typedef MOnNameType AttributeRangesNameType;
typedef MOnNameType GenericTransportTTPNameType;
typedef MOnNameType ManagedElementR2NameType;

/**
Excepciones para los lotes condicionales
*/
exception NONeAliasPackage {};
exception NOPortIdPackage {};

/**

```

10.2 Interfaces – de granularidad fina

*/

/**

10.2.1 AttributeRanges

La clase AttributeRanges permite al sistema gestionado generar informes sobre los valores mínimos y máximos que acepta un determinado atributo, así como sobre la granularidad o incrementos por escalones, de la gama. Cada ejemplar AttributeRanges contiene gamas para los atributos que pertenecen a una clase de objeto. El atributo "kind" en el AttributeRanges denota la clase de objeto para la que se definen las gamas. El "attributeName" especifica el nombre del atributo para el que se define la gama. Así, la gama se define utilizando los atributos "minimum", "maximum", y "granularity" (la "granularity" no está disponible para los tipos float, puesto que no es necesaria).

Para cada ejemplar ManagedElement que representa un elemento de red, se pueden crear uno o más ejemplares AttributeRanges. Existe una relación de contenedencia entre los ejemplares AttributeRanges y el ejemplar ManagedElement.

Las gamas se definen por cada ejemplar ManagedElement, lo que permite que un atributo tenga diferentes gamas dependiendo del elemento de red al que pertenezca. En otras palabras, el alcance de cada ejemplar AttributeRanges son los objetos pertinentes asociados con el ManagedElement que contiene el ejemplar AttributeRanges.

*/

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

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

/**

Se utiliza esta operación para obtener la clase de objeto para la que el ejemplar AttributeRanges está definiendo gamas de atributo. El valor devuelto es una cadena que contiene el nombre de una clase de objeto.

*/

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

/**

Esta operación se utiliza para obtener un conjunto de gamas de atributo para la clase definida en el atributo modalidad. El valor devuelto es un conjunto de estructuras AttributeRangesType, conteniendo cada una de ellas un nombre de atributo, los límites mínimos y máximos, así como la granularidad o incrementos por escalones, permitidos dentro de los límites.

*/

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

```

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

}; // interfaz AttributeRanges

/**
Fábrica AttributeRanges
Este objeto debe crearse al inicializar el sistema gestionado.
*/

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

}; // interfaz AttributeRangesFactory

```

```
/**
```

10.2.2 Transporte genérico TTP

```

*/
/**
Este objeto se utiliza para representar un inventario de puerto físico.
*/

valuetype GenericTransportTTPValueType: truncatable
itut_m3120::NetworkTPValueType
{
/** GenericTransportTTPValueType utiliza la forma heredada de
itut_m3120:: TPValueType:
    public MOnameSetType                supportedByObjectList;
        // apunta al conjunto de circuitos de soporte
        // GET
    public OperationalStateType        operationalState;
        // condicional, está presente si el ejemplar lo soporta.
        // GET
    public AlarmStatusType              alarmStatus;
// condicional, está presente cuando TP soporta comunicaciones
// de notificación de alarmas.
        // GET
    public CurrentProblemSetType        currentProblemList;
        // condicional, está presente cuando TP soporta comunicaciones
        // de notificación de alarmas.
        // GET
    public AlarmSeverityAssignmentProfileNameType
        alarmSeverityAssignmentProfilePointer;
        // condicional, está presente si el ejemplar soporta
// la configuración de la gravedad de la alarma.
        // GET-REPLACE

```

```

GenericTransportTTPValueType uses the following inherited form
itut_m3120:: NetworkTPValueType:
    public PointDirectionalityType pointDirectionality;
        // GET
    public SignalIdType          signalId;
        // GET, SET-BY-CREATE

*/

    public PortIDType PortID;
        // conditional
        // PortIdPackage
        // presente si está representado el puerto TTP del servidor
        // GET

    public MOnameSetType clientLinkEndPointerList;
        // GET-REPLACE

    public PointCapacityType potentialCapacity;
        // condicional
        // presente si el TTP es de tecnología adaptativa de velocidad
        // GET

}; // valuetype GenericTransportTTPValueType

```

```

/**
Definición de la interfaz de granularidad fina
*/

```

```

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)

}; // interfaz GenericTransportTTP

interface GenericTransportTTPFactory:
    itut_x780::ManagedObjectFactory
{
    itut_x780::ManagedObject create
        (in NameBindingType nameBinding,
         // Nombre del módulo que contiene la información
         // de vinculación del nombre.
         in MONameType superior,
         // Nombre del objeto que lo contiene.
         in string reqID,
         // Valor ID solicitado para el nombre;
         // vacío si se utiliza autodenominación.
         out MONameType name,
         // Nombre completo de un objeto recién creado.
         in StringSetType packageNameList,
         // Lista de lotes solicitados.

         in MONameSetType supportedByObjectList,
         // puede ser 0
         in AlarmSeverityAssignmentProfileNameType
         alarmSeverityAssignmentProfilePointer,
in PointDirectionalityType pointDirectionality,

         in MONameSetType clientLinkEndPointList)

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

}; // interfaz GenericTransportTTPFactory

/**

```

10.2.3 ManagedElementR2

Los objetos ManagedElementR2 son objetos gestionados que representan equipos de telecomunicaciones o entidades RGT (grupos o partes) internos a la red de telecomunicaciones que efectúan funciones de elemento gestionado, es decir, proporcionan soporte y/o servicio al suscriptor. Los elementos gestionados pueden o no efectuar adicionalmente funciones de mediación/OS. Un elemento gestionado comunica con el gestor mediante una interfaz CORBA normal a fin de ser supervisado y/o controlado. Un elemento gestionado contiene equipos que pueden o no estar distribuidos geográficamente.

Cuando el objeto Managed Element soporta notificaciones de cambio de valor de atributo, la notificación attributeValueChange será emitida cada vez que cambie el valor de uno de los siguientes atributos: estado de alarma, etiqueta de usuario, versión, nombre de ubicación, lista de problemas actuales y activar alarma local visual audible. Para los atributos precedentes que no pueden ser soportados, el comportamiento vinculado a la emisión de una notificación de cambio de valor de atributo sólo será aplicable cuando el objeto gestionado soporta el atributo. Cuando el objeto soporta notificaciones de cambio de estado, será emitida la stateChangeNotification si cambia el valor de estado administrativo, de estado operacional, o de estado de utilización.

No se permite la supresión del protocolo de gestión. (El objeto deberá emitir la excepción DeleteNotAllowed como respuesta a cualquier operación de supresión.)

Esta interfaz se basa en el objeto Managed Element R2 de la Enmienda 6/M.3100. Se omite aquí la referencia al ManagedElementR1 para armonizar los nombres de los objetos M.3100 y M.3120.

Este valuetype se utiliza para extraer todos los atributos del ManagedElementR2 en una sola operación. La mayor parte de los atributos no soportados se devolverán en forma de cadena vacía o lista. No obstante, la recepción de un valor de cadena vacía no significa que el atributo no es soportado.

*/

```
    valuetype ManagedElementR2ValueType: truncatable
itut_m3120::ManagedElementValueType
{
    public Istring          modelCode;
        // GET
    public IstringSetType   neAliases;
        // conditional
        // neAliasPackage
        // GET
    public MeTypeSetType    managedElementType;
        // GET

}; // tipo de valor ManagedElementR2ValueType

interface ManagedElementR2: itut_m3120::ManagedElement
{
/**
El siguiente método es el código modelo de producto de un elemento de red. El
código modelo de producto es la identificación del modelo del fabricante. Es la
información que presenta el proveedor y que él mismo utiliza para distinguir el
elemento de red entre una familia de productos. Este atributo es utilizado por
el OSS para realizar procesos de detección e inventario de equipo.
*/
    Istring modelCodeGet ()
        raises (itut_x780::ApplicationError);
/**
El siguiente método devuelve una serie de cadenas que contienen alias de
productos del elemento gestionado, según se definen en el EMS. El EMS determina
estos alias para un determinado ejemplar ManagedElement. Disponer de estos alias
vía la interfaz EMS/NMS resulta útil para relacionar los nombres de los
elementos de red introducidos en el EMS, a través de una interfaz de usuario
gráfica, o por cualquier otro medio, con los alias que se encuentran en la
interfaz de usuario NMS. Lo que es más importante, estos alias pueden aparecer
en las alarmas enviadas por determinados programas del EMS a través de una
interfaz distinta de CORBA.
*/
    Istring neAliasesGet ()
        raises (itut_x780::ApplicationError,
            NOneAliasPackage);
/**
El siguiente método devuelve un conjunto de cadenas textuales y UID que describe
un tipo genérico de elemento de red modelado por el ejemplar ManagedElementR2.
Pueden utilizarse múltiples valores de tipo de elemento gestionado para
describir un equipo híbrido.
*/
    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)

}; // interfaz ManagedElementR2

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

```

```

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

}; // interfaz ManagedElementR2Factory

```

/**

10.3 Interfaces fachada

El comportamiento de las interfaces fachada es idéntico al de las correspondientes interfaces de granularidad fina, por lo que no se describe aquí el comportamiento de las interfaces fachada. Los lectores pueden referirse a la cláusula 10.2 en que se describen las interfaces de granularidad fina para conocer el comportamiento de las interfaces fachada.

Esta cláusula puede omitirse del IDL si el sistema de gestión soporta únicamente interfaces de granularidad fina.

*/

/**

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)
}; // interfaz AttributeRanges_F

```

/**

10.3.2 GenericTransportTTP_F

*/

/**

Definición de interfaz de granularidad gruesa

*/

```

interface GenericTransportTTP_F: itut_m3120::NetworkTP_F
{

/**
Los ejemplares GenericTransportTTP se crean utilizando el
GenericTransportTTPFactory o son creados de manera automática por el sistema
gestionado.
*/

/** GenericTransportTTP_F inherits the following methods from
itut_m3120::TP_F:
supportedByObjectListGet,
operationalStateGet, alarmStatusGet, containedInSubnetworkListGet,
currentProblemListGet, alarmSeverityAssignmentProfilePointerGet,
alarmSeverityAssignmentProfilePointerSet

GenericTransportTTP_F inherits the following methods from
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);

/**

Ofrece anchura de banda potencial para las tecnologías de servidor adaptativas
de velocidad.
*/

    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)

}; // interfaz 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)  
  
}; // interfaz ManagedElementR2_F  
  
/**
```

10.4 Vinculación de nombre

```
*/  
  
/**  
El siguiente módulo contiene información sobre la vinculación de nombre.  
*/
```

```
    module NameBinding  
    {
```

```
/**
```

10.4.1 AttributeRanges

```
*/  
  
módulo 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";
}; // módulo AttributeRanges_ManagedElement
```

```
/**
```

10.4.2 GenericTransportTTP

```
*/
```

```
módulo 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";
}; // módulo GenericTransportTTP_ManagedElement
```

```
}; // módulo NameBinding
```

```
}; // módulo itut_m3120
```

```
#endif // _itut_m3120_amd2_idl_
```


SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedia
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedia
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	Redes de datos y comunicación entre sistemas abiertos
Serie Y	Infraestructura mundial de la información y aspectos del protocolo Internet
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación