



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

X.780.1

(08/2001)

SERIE X: REDES DE DATOS Y COMUNICACIÓN
ENTRE SISTEMAS ABIERTOS

Gestión de interconexión de sistemas abiertos –
Funciones de gestión y funciones de arquitectura de
gestión distribuida abierta

**Directrices de la RGT para la definición de
interfaces de objetos gestionados mediante
arquitectura de intermediario de petición de
objeto común de granularidad gruesa**

Recomendación UIT-T X.780.1

RECOMENDACIONES UIT-T DE LA SERIE X
REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS

REDES PÚBLICAS DE DATOS	
Servicios y facilidades	X.1–X.19
Interfaces	X.20–X.49
Transmisión, señalización y conmutación	X.50–X.89
Aspectos de redes	X.90–X.149
Mantenimiento	X.150–X.179
Disposiciones administrativas	X.180–X.199
INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Modelo y notación	X.200–X.209
Definiciones de los servicios	X.210–X.219
Especificaciones de los protocolos en modo conexión	X.220–X.229
Especificaciones de los protocolos en modo sin conexión	X.230–X.239
Formularios para declaraciones de conformidad de implementación de protocolo	X.240–X.259
Identificación de protocolos	X.260–X.269
Protocolos de seguridad	X.270–X.279
Objetos gestionados de capa	X.280–X.289
Pruebas de conformidad	X.290–X.299
INTERFUNCIONAMIENTO ENTRE REDES	
Generalidades	X.300–X.349
Sistemas de transmisión de datos por satélite	X.350–X.369
Redes basadas en el protocolo Internet	X.370–X.399
SISTEMAS DE TRATAMIENTO DE MENSAJES	X.400–X.499
DIRECTORIO	X.500–X.599
GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS	
Gestión de redes	X.600–X.629
Eficacia	X.630–X.639
Calidad de servicio	X.640–X.649
Denominación, direccionamiento y registro	X.650–X.679
Notación de sintaxis abstracta uno	X.680–X.699
GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Marco y arquitectura de la gestión de sistemas	X.700–X.709
Servicio y protocolo de comunicación de gestión	X.710–X.719
Estructura de la información de gestión	X.720–X.729
Funciones de gestión y funciones de arquitectura de gestión distribuida abierta	X.730–X.799
SEGURIDAD	X.800–X.849
APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Compromiso, concurrencia y recuperación	X.850–X.859
Procesamiento de transacciones	X.860–X.879
Operaciones a distancia	X.880–X.899
PROCESAMIENTO DISTRIBUIDO ABIERTO	X.900–X.999

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

Recomendación UIT-T X.780.1

Directrices de la RGT para la definición de interfaces de objetos gestionados mediante arquitectura de intermediario de petición de objeto común de granularidad gruesa

Resumen

Esta Recomendación define extensiones al conjunto de directrices para el modelado de objetos gestionados CORBA, de la red de gestión de las telecomunicaciones (RGT), requeridas para el soporte de interfaces de granularidad gruesa. Proporciona también directrices para la conversión de interfaces de granularidad fina en interfaces de granularidad gruesa. Se proporciona un módulo CORBA IDL que define tipos de interfaz de base que habrán de ser extendidos.

Orígenes

La Recomendación UIT-T X.780.1, 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 13 de agosto de 2001.

Palabras clave

Arquitectura de intermediario de petición de objeto común (CORBA, *common object request broker architecture*), directrices para la definición de objetos gestionados (GDMO, *guidelines for the definition of managed objects*), lenguaje de definición de interfaz (IDL, *interface definition language*), procesamiento distribuido, interfaces de la RGT, objetos gestionados.

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 2002

Es propiedad. Ninguna parte de esta publicación puede reproducirse o utilizarse, de ninguna forma o por ningún medio, sea éste electrónico o mecánico, de fotocopia o de microfilm, sin previa autorización escrita por parte de la UIT.

ÍNDICE

Página

1	Alcance	1
1.1	Finalidad	1
1.2	Aplicación.....	1
1.3	Estructura de la Recomendación.....	2
2	Referencias normativas.....	2
3	Definiciones	3
3.1	Definiciones de UIT-T X.701	3
3.2	Definiciones de UIT-T X.703	3
3.3	Otras definiciones	3
4	Abreviaturas.....	3
5	Convenios	4
5.1	Convenios	4
5.2	Compilación del IDL	5
6	Consideraciones sobre el diseño de interfaces de granularidad gruesa	5
6.1	Creación y supresión de objetos de granularidad gruesa.....	5
6.2	Atributos	5
6.3	Notificaciones	5
6.4	Acceso de granularidad gruesa a todos los recursos gestionados	6
6.5	Excepciones	6
6.6	Soporte de todas las operaciones	6
6.7	Correspondencia prescriptiva	6
6.8	Recuperación de atributos desde múltiples objetos	6
7	Visión general del marco y los requisitos.....	6
7.1	Visión general del marco	6
7.2	Visión general de las extensiones de granularidad gruesa.....	7
7.2.1	Esquema de diseño de la fachada	8
7.2.2	Extensión de nombre de objeto gestionado	9
7.2.3	Servicios de soporte para objetos gestionados accesibles a través de una fachada.....	9
7.2.4	Modelado de fachada.....	10
8	Aprovisionamiento de interfaces de fachada para acceder a objetos gestionados.....	11
8.1	Ejemplificación de fachada.....	11
8.2	Clase de base de interfaz de fachada	12
8.2.1	Capacidades básicas de fachada de objeto gestionado	12

	Página
8.2.2 IDL de fachada de objeto gestionado	12
8.2.3 Operación <i>objectClassGet()</i>	13
8.2.4 Operación <i>packagesGet()</i>	13
8.2.5 Operación <i>creationSourceGet()</i>	13
8.2.6 Operación <i>deletePolicyGet()</i>	14
8.2.7 Operación <i>attributesGet()</i>	14
8.2.8 Operación <i>attributesBulkGet()</i>	14
8.2.9 Operación <i>destroy()</i>	16
8.3 Interfaz iteradora <i>AttributesBulkGet</i>	16
8.4 Ejemplificación de fábricas	17
9 Directrices para el modelado CORBA de granularidad gruesa	18
10 Directrices para la traducción de modelos de granularidad fina en modelos de granularidad gruesa.....	18
11 Cumplimiento y conformidad con respecto al IDL de granularidad gruesa.....	19
11.1 Cumplimiento por parte de documentos relativos a normas.....	19
11.2 Conformidad de sistema	20
11.3 Directrices para los enunciados de conformidad	20
Anexo A – IDL de modulado de granularidad gruesa	20

Recomendación UIT-T X.780.1

Directrices de la RGT para la definición de interfaces de objetos gestionados mediante arquitectura de intermediario de petición de objeto común de granularidad gruesa

1 Alcance

La arquitectura de la red de gestión de telecomunicaciones (RGT) definida en UIT-T M.3010 (2000) introduce conceptos de procesamiento distribuido e incluye la utilización de múltiples protocolos de gestión. Las Recomendaciones UIT-T Q.816 y X.780 definen subsiguientemente, dentro de esta arquitectura, un marco para aplicar la arquitectura de intermediario de petición de objeto común (CORBA, *common object request broker architecture*) como uno de los protocolos de gestión de la RGT.

Esta Recomendación, junto con UIT-T Q.816.1, completan este marco con especificaciones que permiten el soporte de un estilo de interacción, entre sistemas de gestión y sistemas gestionados, algo diferente del que fue especificado en los documentos del marco original. Este estilo de interacción tiene algunas ventajas; la principal es que el sistema de gestión ya no tiene que recuperar una dirección lógica orientada a objeto para cada uno de los recursos gestionables a que desea ganar acceso. Los grandes sistemas podrían tener millones de estas direcciones lógicas. También hay algunos cambios en la estructuración del soporte lógico en los sistemas gestionados, con las consiguientes ventajas para algunos proveedores de estos sistemas.

Esta Recomendación tiene el mismo alcance que el marco original de la RGT basada en la arquitectura CORBA. El marco y estas extensiones abarcan todas las interfaces de la RGT que permiten el uso CORBA. Sin embargo, no cabe esperar que todas las capacidades y servicios definidos aquí sean necesarios en todas las interfaces de la RGT. Esto implica que el marco se puede utilizar para interfaces entre sistemas de gestión en todos los niveles de abstracción (dentro de una administración o entre administraciones), así como también entre sistemas de gestión y elementos de red.

1.1 Finalidad

Esta Recomendación tiene por finalidad ampliar el marco CORBA de la RGT para que sea posible utilizarlo en una más amplia gama de aplicaciones. Las extensiones permiten un modo de interacción algo diferente entre el sistema de gestión y el sistema gestionado, lo que es preferible en muchas situaciones. Por tanto, esta Recomendación está destinada a los diversos grupos que se ocupan de la especificación de interfaces de gestión de red.

1.2 Aplicación

En las Recomendaciones sobre el marco CORBA de la RGT, los recursos gestionables de la red se modelan como objetos lógicos a los que se puede acceder utilizando CORBA. Modelos de información escritos en el lenguaje de definición de interfaz (IDL, *interface definition language*) CORBA describen las interfaces de objetos.

CORBA proporciona transparencia con respecto al emplazamiento, de forma que un objeto lógico puede interactuar con otro cualquiera que sea su emplazamiento. Para el acceso a un objeto lógico se utiliza lo que en CORBA se denomina una referencia de objeto interoperable (IOR, *interoperable object reference*).

El marco CORBA original de la RGT modela cada recurso gestionable como un objeto CORBA independiente, identificado con una IOR única. Este método flexible permite que cada objeto resida en cualquier lugar. Sin embargo, es necesario que los sistemas de gestión dispongan de una IOR para cada uno de los objetos a que desean acceder. Muchas empresas y administraciones en la industria de las telecomunicaciones buscan soluciones para evitar este laborioso proceso. Por otra parte, en algunos casos también será necesario que el sistema gestionado soporte un gran número de estas IOR, lo que tratan de evitar algunos proveedores de estos sistemas. Esta Recomendación junto con UIT-T Q.816.1, define la forma en que el marco CORBA de la RGT debe definirse para evitar la utilización de un gran número de referencias IOR.

Las interfaces basadas en CORBA que permiten direccionar cada recurso gestionable con una IOR única se suelen llamar interfaces "de granularidad fina". Otras interfaces en las que no se asigna una IOR a cada recurso gestionable se suelen llamar interfaces "de granularidad gruesa".

Dado que en esta Recomendación se define un enfoque algo diferente para el modelado de recursos gestionables en interfaces de granularidad gruesa, las especificaciones del modelo de interfaz serán algo diferentes, según que se trate de una interfaz de granularidad fina o de granularidad gruesa.

1.3 Estructura de la Recomendación

Esta Recomendación está organizada así:

- Cláusula 1 Introducción, estructura y actualizaciones.
- Cláusula 2 Referencias.
- Cláusulas 3 y 4 Definición de las abreviaturas utilizadas en esta Recomendación.
- Cláusula 5 Convenios.
- Cláusula 6 Aspectos de diseño que deben considerarse para completar el marco con interfaces de granularidad gruesa.
- Cláusula 7 Visión general de los requisitos del marco CORBA de la RGT y de los relativos a la granularidad gruesa.
- Cláusula 8 Aprovisionamiento de interfaces de fachada para el acceso a objetos gestionados. Esta cláusula trata de las interfaces específicas del modelo que es necesario implementar en las interfaces de granularidad gruesa.
- Cláusula 9 Directrices para la definición de interfaces CORBA de granularidad gruesa.
- Cláusula 10 Directrices para la traducción de especificaciones de interfaz CORBA de granularidad fina a especificaciones de interfaz de granularidad gruesa.
- Cláusula 11 Directrices de cumplimiento y conformidad.
- Anexo A IDL de modulado de granularidad gruesa. Este anexo es normativo.

2 Referencias normativas

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 aplicar 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.

- [1] UIT-T X.780 (2001), *Directrices de la RGT para la definición de objetos gestionados mediante arquitectura de intermediario de petición de objeto común*.

- [2] UIT-T Q.816 (2001), *Servicios de la RGT basados en arquitectura de intermediario de petición de objeto común.*
- [3] 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.*
- [4] OMG Document formal/99-10-07, *The Common Object Request Broker: Architecture and Specification*, Revision 2.3.1.

3 Definiciones

3.1 Definiciones de UIT-T X.701

Los siguientes términos utilizados en esta Recomendación están definidos en Visión general de la gestión de sistemas (UIT-T X.701):

- clase de objeto gestionado;
- gestor;
- agente.

3.2 Definiciones de UIT-T X.703

El siguiente término utilizado en esta Recomendación está definido en Arquitectura de gestión distribuida abierta (UIT-T X.703):

- notificación.

3.3 Otras definiciones

3.3.1 fachada: Una interfaz de objeto definida para proporcionar acceso a un conjunto de objetos gestionados, todos los cuales son de la misma clase.

4 Abreviaturas

En esta Recomendación se utilizan las siguientes siglas.

CMIP	Protocolo común de información de gestión (<i>common management information protocol</i>)
CORBA	Arquitectura de intermediario de petición de objeto común (<i>common object request broker architecture</i>)
COS	Servicios de objeto comunes (<i>common object services</i>)
DN	Nombre distinguido (<i>distinguished name</i>)
EMS	Sistema de gestión de elementos (<i>element management system</i>)
GDMO	Directrices para la definición de objetos gestionados (<i>guidelines for the definition of managed objects</i>)
ID	Identificador
IDL	Lenguaje de definición de interfaz (<i>interface definition language</i>)
IOP	Protocolo de interoperabilidad de Internet (<i>Internet interoperability protocol</i>)
IOR	Referencia de objeto interoperable (<i>interoperable object reference</i>)
MO	Objeto gestionado (<i>managed object</i>)

NE	Elemento de red (<i>network element</i>)
NMS	Sistema de gestión de red (<i>network management system</i>)
OAM&P	Operación, administración, mantenimiento y aprovisionamiento (<i>operations, administration, maintenance, and provisioning</i>)
OID	Identificador de objeto (<i>object identifier</i>)
OMG	Grupo de gestión de objetos (<i>object management group</i>)
ORB	Mediador de petición de objeto (<i>object request broker</i>)
OSI	Interconexión de sistemas abiertos (<i>open systems interconnection</i>)
PDU	Unidad de datos de protocolo (<i>protocol data unit</i>)
POA	Adaptador de objeto portátil (<i>portable object adapter</i>)
QoS	Calidad de servicio (<i>quality of service</i>)
RDN	Nombre distinguido relativo (<i>relative distinguished name</i>)
RGT	Red de gestión de las telecomunicaciones
UID	Identificador universal (<i>universal identifier</i>)
UIT-T	Unión Internacional de Telecomunicaciones – Sector de Normalización de las Telecomunicaciones
UML	Lenguaje de modelado unificado (<i>unified modelling language</i>)
UTC	Tiempo universal coordinado (<i>universal time coordinated</i>)

5 Convenios

5.1 Convenios

Se siguen algunos convenios en esta Recomendación para que el lector se dé cuenta de la finalidad del texto. Mientras que la mayor parte de esta Recomendación es de carácter normativo, los párrafos que sucintamente indican requisitos que deben ser cumplidos obligatoriamente por un sistema de gestión (sistema gestor o gestionado) están precedidos por una "R" en negrita encerrada entre paréntesis, seguida de un nombre corto que indica a qué se aplica el requisito, y de un número. Por ejemplo:

(R) EJEMPLO-1 Un requisito obligatorio relativo a un ejemplo.

Los requisitos que pueden ser implementados facultativamente por un sistema de gestión están precedidos por una "O" (*optional*, facultativo, opcional) en lugar de una "R". Por ejemplo:

(O) OPCIÓN-1 Un requisito facultativo relativo a un ejemplo.

Los enunciados de requisitos se utilizan para crear perfiles de cumplimiento y conformidad.

En esta Recomendación se incluyen muchos ejemplos del IDL de CORBA (brevemente, CORBA IDL), y en el anexo A se incluye IDL que especifica los servicios concretos de la RGT y los tipos de datos de soporte. El IDL está escrito con tipo de letra courier de 9 puntos.

```
// Example IDL
interface foo {
    void operation1 ();
};
```

5.2 Compilación del IDL

Una ventaja de utilizar el IDL para especificar interfaces de gestión de red es que el IDL puede ser "compilado" en un código de programación por herramientas que acompañan a un intermediario de petición de objetos (ORB). De esta forma se automatiza la elaboración de una parte del código necesario para que puedan interfundar las aplicaciones de gestión de red. El código contenido en el anexo A puede ser de interés para implementadores, que podrán extraerlo y compilarlo. El anexo A es normativo y debe ser utilizado por los responsables de la implementación de sistemas conformes con esta Recomendación. El IDL presentado en esta Recomendación ha sido verificado en dos compiladores para una mayor seguridad de su funcionamiento. Se debe utilizar un compilador que soporte la versión CORBA especificada en UIT-T Q.816.

El formato del anexo A facilita la operación de cortar y pegar en ficheros de texto simple, que luego pueden ser compilados. A continuación se dan algunas indicaciones sobre la forma de realizar esta operación.

- 1) Al parecer, la operación de cortar y pegar funciona mejor cuando se utiliza la versión Microsoft® Word® de esta Recomendación. Parece que la operación de cortar y pegar cuando se utiliza el formato Adobe® Acrobat® incluye encabezamientos y notas de pie de página, que no pueden ser compilados.
- 2) Todo el anexo A, desde la línea `"/* This IDL code ..."` hasta el final se debe almacenar en un fichero con el nombre `"itut_x780_1.idl"` en un directorio donde lo pueda encontrar el compilador IDL.
- 3) No es necesario retirar los títulos insertados en el anexo A. El compilador no los tendrá en cuenta porque están encapsulados en comentarios IDL.
- 4) Los compiladores que convierten el lenguaje IDL en HTML reconocen los comentarios que empiezan con la secuencia especial `"/**"`. Muchos de estos comentarios contienen instrucciones de formato especiales para estos compiladores. Cuando se trabaja con IDL, también puede ser interesante generar HTML, porque los ficheros HTML resultantes tienen enlaces que facilitan la navegación en los ficheros.
- 5) El formato del anexo A con tabuladores a intervalos de ocho espacios y nuevas líneas insertadas expresamente, permitirá normalmente el tratamiento con cualquier editor de texto.

6 Consideraciones sobre el diseño de interfaces de granularidad gruesa

En esta cláusula se estudian varios aspectos de diseño que se deben tener en cuenta en el marco con miras al soporte de interfaces de granularidad gruesa.

6.1 Creación y supresión de objetos de granularidad gruesa

Será posible crear y suprimir representaciones de granularidad gruesa de los recursos gestionados. Se debe investigar la posibilidad de incluir la operación de creación en la interfaz.

6.2 Atributos

El marco soportará la asociación de atributos con recursos gestionados a los que se gana acceso a través de una interfaz.

6.3 Notificaciones

El marco soportará notificaciones de eventos desde recursos gestionados a los que se gana acceso a través de interfaces de granularidad gruesa.

6.4 Acceso de granularidad gruesa a todos los recursos gestionados

El marco hará posible, y requerirá, que las implementaciones permitan a los sistemas de gestión ganar acceso, a todos los recursos gestionados a través de interfaces de granularidad gruesa.

6.5 Excepciones

Las interfaces de granularidad gruesa permitirán que los recursos gestionados formulen una excepción cuando se invoca una operación. Las excepciones para cada operación deben estar definidas explícitamente.

6.6 Soporte de todas las operaciones

Una interfaz de granularidad gruesa soportará todas las operaciones aplicables a un recurso gestionado.

6.7 Correspondencia prescriptiva

La correspondencia entre los modelos de información de granularidad fina y los modelos de información de granularidad gruesa debe ser prescriptiva. Es indispensable que la correspondencia se pueda establecer por algoritmo. Si un modelo de información IDL se desarrolla traduciendo un modelo de información GDMO, toda optimización efectuada manualmente durante la traducción aparecerá tanto en el modelo de granularidad fina como en el de granularidad gruesa.

6.8 Recuperación de atributos desde múltiples objetos

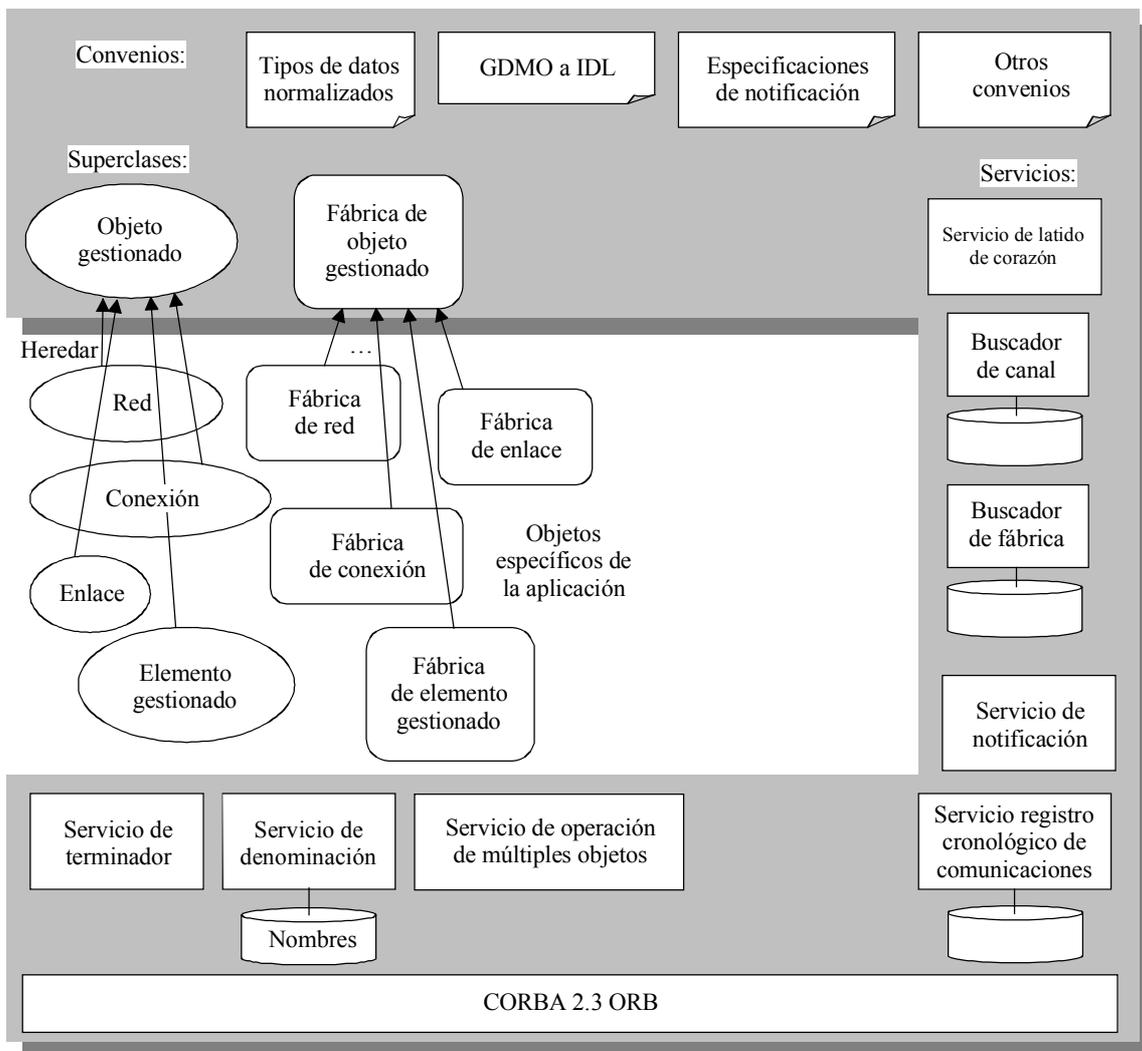
Es necesario recuperar atributos de múltiples ejemplares de objetos gestionados del mismo tipo en una sola operación sumamente tipificada. Como todos estos objetos van a ser tratados utilizando la misma fachada, esto se puede conseguir con una sola operación realizada por la fachada.

7 Visión general del marco y los requisitos

En la cláusula 6 se examinaron los aspectos de diseño que deben resolverse para la inclusión del soporte de las interfaces de granularidad gruesa en el marco. En esta cláusula y en el resto de esta Recomendación se explica en detalle cómo se ha de extender el marco para tener en cuenta estos aspectos. La Recomendación UIT-T Q.816.1 trata específicamente los servicios de soporte del marco para interfaces de granularidad gruesa, mientras que en la presente Recomendación se definen directrices para desarrollar modelos de información para las interfaces de granularidad gruesa. Tras una breve descripción del marco actual se describirán las extensiones.

7.1 Visión general del marco

El marco para interfaces de la RGT basadas en CORBA consiste en una serie de capacidades. Un elemento central del marco es un conjunto de servicios de objeto comunes OMG. El marco define el papel que desempeña en las interfaces de gestión de red, y convenios para su utilización. El marco también define servicios de soporte que no han sido normalizados como servicios de objeto comunes OMG, pero que cabe esperar serán normalizados en las interfaces de gestión de red conformes con el marco.



T0415630-01

Figura 1/X.780.1 – Visión general del marco

El marco se representa en la figura 1. Aparece en gris, y en el centro están representados los objetos específicos de la aplicación soportados por el marco. La casilla en la parte inferior representa el mediador de petición de objetos (ORB, *object request broker*) basado en CORBA. Por encima de esta casilla hay varias casillas con nombres que representan los servicios que constituyen el marco. (En algunos aparecen iconos para señalar las bases de datos necesarias para realizar sus funciones.) La Recomendación UIT-T Q.816 define estos servicios y los requisitos de la versión ORB. Los iconos en la parte superior de la figura representan dos superclases, una para objetos gestionados y la otra para fábricas de objetos gestionados. Cada uno de los objetos gestionados y de las fábricas de objetos gestionados debe heredar de estas dos superclases, respectivamente. Los iconos de páginas con una esquina doblada representan convenios de modelado de objetos normalizados. La Recomendación UIT-T X.780 define estos convenios y las superclases.

7.2 Visión general de las extensiones de granularidad gruesa

En esta cláusula se describen someramente las extensiones del marco que se requieren para el soporte de interfaces de granularidad gruesa.

7.2.1 Esquema de diseño de la fachada

La principal modificación del marco requerido para el soporte de interfaces de granularidad gruesa es la forma de acceso a los objetos gestionados. El número de objetos gestionados en un sistema gestionado tiene que poder aumentar sin que aumente el número de IOR soportados por el sistema. Ahora bien, se sigue prefiriendo que el acceso a los objetos gestionados sea sumamente tipificado. Con este fin se ha definido el esquema de diseño designado aquí por el esquema "de fachada". Puede considerarse que una fachada es como un falso frente o un portal. Utilizando el esquema de diseño de fachada, un sistema gestionado soportará un número reducido de interfaces de fachada, al menos una y generalmente unas cuantas, para cada tipo de objeto gestionado en el sistema. Para invocar una operación en un objeto gestionado, un sistema de gestión en realidad invoca la operación en una fachada para ese tipo de objeto gestionado en ese sistema. En el esquema de diseño de fachada, no es necesario que los objetos gestionados expongan una interfaz CORBA y, por tanto, pueden no tener IOR individuales. De esta forma, un sistema gestionado que soporta el método de fachada no tendrá que utilizar interfaces de objetos gestionados de granularidad fina.

Conviene formarse la idea de que una fachada no es un objeto gestionado, sino más bien un objeto intermediario que permite a un sistema de gestión acceder a objetos gestionados. El objeto fachada tiene una interfaz CORBA y es accesible mediante CORBA. Sin embargo, es posible que los objetos gestionados no tengan interfaces CORBA y que no puedan ser accedidos directamente mediante CORBA. La fachada propiamente dicha no representa un recurso de red gestionable; tiene por finalidad permitir la interacción con los objetos que sí representan recursos gestionables. El sistema gestionado crea automáticamente todos los objetos fachada, los cuales existirán mientras los objetos gestionados sean accesibles a través de esa fachada. Una interfaz de granularidad gruesa puede tener varias fachadas para el mismo tipo de objetos gestionados, pero un determinado objeto gestionado sólo podrá ser accesible a través de una sola fachada. Véase la figura 2.

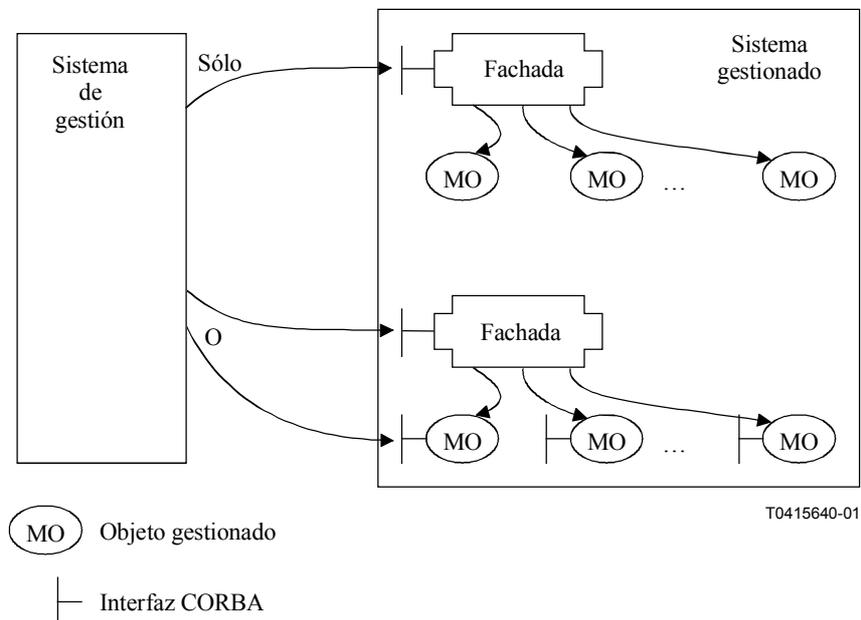


Figura 2/X.780.1 – Papel desempeñado por la fachada

La figura representa un sistema de gestión y que gana acceso a un sistema gestionado que trabaja según el método de granularidad gruesa. El sistema gestionado tiene dos interfaces de fachada que permiten al sistema de gestión ganar acceso a dos conjuntos diferentes de objetos gestionados. A los objetos gestionados representados en la parte superior de la figura sólo se puede ganar acceso a través de la fachada. Los objetos gestionados representados en la parte inferior también soportan

interfaces CORBA directas; el acceso es posible a través de la fachada o directamente. El acceso CORBA directo es facultativo, pero un sistema gestionado que emplea el método de fachada proporcionará interfaces de fachada para todos los ejemplares de objetos gestionados.

Una fachada puede utilizar una interfaz CORBA de objeto gestionado para invocar una operación sobre ese objeto, o algún otro medio específico de la implementación. De hecho, un sistema gestionado no necesita ni siquiera implementar objetos gestionados como objetos individuales, internamente. Ahora bien, implementando una interfaz basada en este marco, da la impresión de que los objetos gestionados son objetos implementados internamente.

Cuando se invoca una operación sobre un objeto gestionado a través de una fachada, la fachada invocará la operación sobre el objeto gestionado o la entidad reales. Puesto que una sola fachada permite el acceso a muchos objetos gestionados, la fachada debe saber a cuál de estos objetos gestionados está destinada realmente la operación. Por eso se adopta el convenio de incluir el nombre del objeto gestionado de destino como el primer parámetro de cada operación de la fachada dirigida a un objeto gestionado.

Si bien es posible que los objetos gestionados ya no tengan IOR únicas, tendrán todavía nombres únicos, y se puede seguir considerándolos como entidades individuales que representan recursos gestionables.

7.2.2 Extensión de nombre de objeto gestionado

Ya se ha dicho que los objetos gestionados accesibles a través de una fachada siguen teniendo un nombre, aunque puedan no tener una interfaz CORBA individual. Es importante que un sistema de gestión pueda determinar, basándose en el nombre del objeto gestionado, la fachada que habrá de utilizar. De lo contrario, tendría que interrogar el sistema gestionado o asociar de forma permanente una IOR de fachada con cada nombre de objeto gestionado. Para poder determinar una fachada de objeto gestionado basándose únicamente en su nombre, los nombres de objetos gestionados accesibles a través de la fachada se alargan algo más en comparación con los nombres de objetos gestionados no accesibles a través de la fachada. En el componente de nombre final, que tiene siempre una cadena ID con el valor "Object" (o <empty>, según la extensión definida por UIT-T Q.816.1, la cadena *kind* se fija al valor de un identificador de fachada asignado a la fachada que permite acceder al objeto. La cadena *kind* está vacía si los objetos gestionados no son accesibles a través de una fachada. La Recomendación UIT-T Q.816.1 da más detalles sobre la utilización de la cadena *kind* en el componente de nombre de objeto gestionado final para identificar una fachada.

7.2.3 Servicios de soporte para objetos gestionados accesibles a través de una fachada

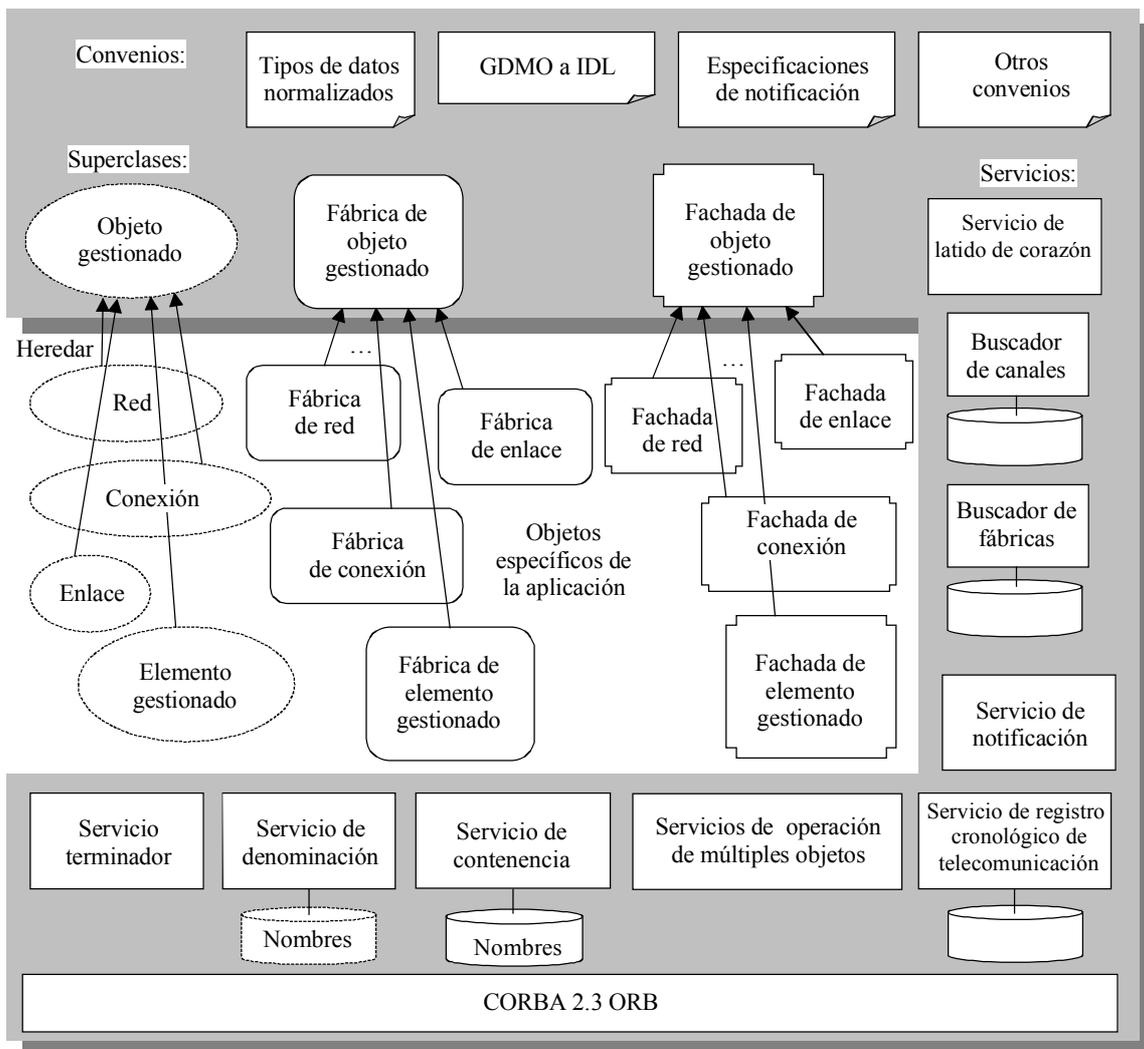
Los servicios de soporte del marco proporcionados en interfaces que utilizan el método de fachada son en gran parte los mismos definidos en UIT-T Q.816. Algunos servicios como el buscador de fábrica y buscador de canal, no requieren modificación. Otros, como el servicio terminador o el servicio de operación de múltiples objetos (MOO, *multiple object operation*), no necesitan ninguna modificación de sus interfaces en lo que concierne a la forma de utilización por los sistemas de gestión, pero sí podrían ser necesarios algunos cambios en sus implementaciones si acceden a objetos gestionados a través de interfaces de fachada de objetos gestionados (y no por otro método específico de la implementación). La Recomendación UIT-T Q.816.1 especifica las modificaciones necesarias en los servicios de soporte de marco para el soporte de interfaces de granularidad gruesa.

La principal modificación de los servicios de soporte se refiere al soporte de la denominación. Las interfaces de fachada están vinculadas a nombres en el servicio de denominación en una forma muy similar al caso de las interfaces de servicio de soporte. Sin embargo, en las interfaces que utilizan fachadas no es necesario que los nombres de los objetos gestionados estén vinculados a IOR en el servicio de denominación OMG. En cambio, se ha introducido un nuevo servicio para almacenar nombres de objetos gestionados e información de relaciones de contención. El nuevo servicio, el servicio de contención, está definido en UIT-T Q.816.1.

7.2.4 Modelado de fachada

Se ha introducido una nueva interfaz de base para soportar el modelo de fachada y la definición de fachadas para utilización con este marco. Se trata de la interfaz que será conocida como interfaz de fachada de objeto gestionado. En las interfaces de granularidad gruesa, desempeña el mismo papel desempeñado por la interfaz de objeto gestionado en las interfaces de granularidad fina. Es la interfaz de base de la que deben heredar, directa o indirectamente, todas las interfaces de fachada de objeto gestionado para funcionar con el marco. La interfaz de fachada de objeto gestionado es muy similar a la interfaz de objeto gestionado definida en UIT-T X.780. Véase la definición de la interfaz de fachada de objeto gestionado en la cláusula 8.

Las modificaciones del marco pueden verse en la figura 3. Se ha añadido una nueva superclase *ManagedObjectFacade*. También se ha añadido el servicio de contenencia. Obsérvese que este servicio permite acceder a una base de datos de nombres de objetos gestionados. La base de datos de nombres de objetos gestionados mantenida por el servicio de denominación se ha representado con líneas de trazo discontinuo para indicar que no necesita almacenar los nombres e IOR de objetos gestionados. Ahora bien, el servicio de denominación sigue siendo necesario para que los sistemas de gestión puedan encontrar las interfaces de fachada y las referencias de los servicios de soporte. Por último, los objetos gestionados también se han representado por líneas de trazo discontinuo para indicar que no tienen que ser accesibles directamente.



T0415650-01

Figura 3/X.780.1 – Marco con extensiones para el soporte de interfaces de granularidad gruesa

8 Aprovisionamiento de interfaces de fachada para acceder a objetos gestionados

Como ya se ha dicho, las interfaces de fachada proporcionan otra forma de acceso a objetos gestionados. En una interfaz de granularidad gruesa (que tiene interfaces de fachada), los objetos gestionados pueden o no ser accesibles mediante interfaces CORBA individuales. Por tanto, un sistema de gestión puede utilizar una fachada para acceder a objetos gestionados invocando operaciones sobre la fachada. Incluso si el sistema gestionado no implementa objetos gestionados como objetos individuales, al soportar esta Recomendación da esa impresión.

8.1 Ejemplificación de fachada

En esta cláusula se definen los requisitos que deben cumplir los sistemas gestionados cuando proporcionan interfaces de fachada para el acceso a objetos gestionados.

(R) FACHADA-1 – Un sistema gestionado proporcionará como mínimo una interfaz de fachada para cada clase de objetos gestionados que se pueda ejemplificar en el sistema, aunque estos objetos también soporten interfaces CORBA directas. No es necesario proporcionar interfaces de fachada para clases de objetos gestionados que no pueden ser ejemplificados en el sistema. Esto incluye las superclases. Por tanto, no es necesario ejemplificar una fachada para una superclase de objetos gestionados si los objetos de este tipo no pueden ser ejemplificados, incluso si objetos gestionados de la subclase pueden ser ejemplificados. Esto no significa, sin embargo que no sea necesario definir interfaces de superclases. La definición de interfaces de fachada sigue la misma jerarquía de herencia que los objetos gestionados. Véanse las siguientes cláusulas 9 y 10. Los requisitos de vinculación de nombres para las interfaces de fachada se definen en UIT-T Q.816.1.

El hecho de que el sistema gestionado no tenga que crear interfaces de fachada para superclases que no son ejemplificables en el sistema no impide aprovechar las ventajas del polimorfismo en un sistema de gestión. Un sistema de gestión puede seguir tratando una interfaz de fachada como una superclase de la fachada y acceder a las capacidades disponibles a través de la superclase, de la misma forma que cualquier cliente CORBA puede tratar un objeto como una superclase. Ahora bien, esta operación será invocada, en realidad, sobre la fachada de subclase, utilizando la IOR de esa fachada. El lenguaje de programación en el sistema cliente se encarga del polimorfismo.

(R) FACHADA-2 – Un sistema gestionado puede proporcionar múltiples interfaces de fachada para una determinada clase de objetos gestionados. Los objetos gestionados accesibles a través de una determinada interfaz de fachada no serán accesibles a través de ninguna otra. La Recomendación UIT-T Q.816.1 explica detalladamente cómo identificar la fachada que habrá de utilizarse para el acceso a un objeto gestionado basándose en su nombre.

(R) FACHADA-3 – El sistema gestionado crea y suprime todas las interfaces de fachada. Una interfaz de fachada debe existir durante todo el tiempo que exista uno de los objetos gestionados accesibles a través de ella. No se envían notificaciones cuando se crea o se suprime una fachada. Por consiguiente, el sistema de gestión se enterará de que hay una nueva fachada si empiezan a aparecer objetos gestionados con nombres que indican que son accesibles a través de la nueva fachada, y únicamente cuando esto ocurra.

(R) FACHADA-4 – Cada operación de fachada destinada a un determinado objeto gestionado contendrá el nombre de ese objeto gestionado en el primer parámetro de la operación. La fachada invocará esa operación sobre el objeto gestionado denominado y retornará los resultados, incluidas las excepciones. Si el objeto gestionado denominado no es accesible a través de esa fachada, la fachada formulará una excepción *applicationError* en respuesta a la operación. El código de error en esta excepción se fijará al código *objectNotFound* definido en el IDL del anexo A.

8.2 Clase de base de interfaz de fachada

En esta cláusula se describe una clase de base fachada de objeto gestionado, definida en el IDL del anexo A. Todas las interfaces de fachada de objeto gestionado proporcionadas en un sistema gestionado deben heredar, directa o indirectamente, de esta interfaz. Esta interfaz proporciona un conjunto de operaciones básicas que todas las interfaces de fachada deben soportar para que puedan utilizarse con este marco.

La interfaz de fachada de objeto gestionado, llamada "*ManagedObject_F*" en el código IDL del anexo A, desempeña el mismo papel en las interfaces de granularidad gruesa que el desempeñado por *ManagedObject* en las interfaces de granularidad fina. Por tanto, es muy similar a la interfaz *ManagedObject* definida en UIT-T X.780. La mayoría de las operaciones de la interfaz *ManagedObject_F* son casi idénticas a las operaciones de la interfaz *ManagedObject*. Se añadió una nueva operación *attributesBulkGet*, y se suprimió la operación *nameGet*.

8.2.1 Capacidades básicas de fachada de objeto gestionado

Todas las interfaces de fachada de objeto gestionado deben soportar las siguientes capacidades:

- Un método que retorna el nombre de clase de un objeto gestionado denominado.
- Un método que retorna los lotes condicionales soportados por un objeto gestionado denominado.
- Un método que retorna la fuente de creación de un objeto gestionado denominado (ya sea creado de forma autónoma por el recurso gestionado, en respuesta a una operación de gestión, o desconocido).
- Un método que retorna la política de supresión para un objeto gestionado denominado. Se trata de un valor enumerado que indica si el objeto no es suprimible, si sólo se puede suprimir cuando no contiene objetos, o si todos los objetos contenidos serán suprimidos cuando se suprima éste.
- Un método que retorna un objeto de tipo de valor CORBA que contiene todos los atributos legibles para el objeto gestionado denominado.
- Un método que retorna un objeto de tipo de valor CORBA que contiene todos los atributos legibles para un conjunto de objetos gestionados.
- Una operación de destrucción.

8.2.2 IDL de fachada de objeto gestionado

La interfaz *ManagedObject_F* se describe (sin comentarios) por el siguiente IDL:

```
interface ManagedObject_F {
    ObjectClassType objectClassGet (in NameType name)
        raises (ApplicationError);

    StringSetType packagesGet (in NameType name)
        raises (ApplicationError);

    SourceIndicatorType creationSourceGet (in NameType name)
        raises (ApplicationError);

    DeletePolicyType deletePolicyGet (in NameType name)
        raises (ApplicationError);

    ManagedObjectValueType attributesGet (
        in      NameType name,
        inout   StringSetType attributeNames)
        raises (ApplicationError);
}
```

```

boolean attributesBulkGet (
    in      NameSetType      names,
    in      StringSetType   attributeNames,
    in      unsigned short   howMany,
    out     AttributesGetResultSet  attributes,
    out     AttributesGetResultIterator  iterator)
    raises  (ApplicationError);

void destroy(in NameType name)
    raises  (ApplicationError,
            DeleteError);

}; // end of ManagedObject_F interface

```

8.2.3 Operación *objectClassGet()*

La operación *objectClassGet()* retorna el nombre de la interfaz (dentro del alcance correspondiente) del objeto gestionado denominado. El objeto gestionado denominado es el objeto gestionado con el nombre incluido en el primer parámetro "name".

Obsérvese que este valor es diferente del nombre de clase de la fachada. Dado que los objetos gestionados en una interfaz de granularidad gruesa también pueden soportar interfaces CORBA directas, se ha decidido requerir que esta operación retorne el mismo valor que obtendría un sistema de gestión que invocara la operación equivalente directamente sobre el objeto gestionado. Si no se ha definido ninguna interfaz de objeto gestionado de granularidad fina para la clase de objetos gestionados accesibles a través de la fachada, la respuesta será el nombre de interfaz dentro del alcance correspondiente de la fachada sin la "_F" final. El mismo valor obtenido en respuesta a esta operación se incluirá en las notificaciones desde el objeto, y en el tipo de valor para el objeto. El tipo de retorno, *ObjectClassType*, es una definición de tipo para cadena.

Dada una referencia a una fachada, un sistema de gestión puede determinar el tipo de fachada de que se trata mediante llamadas CORBA estándar (por ejemplo, la llamada *get_interface* en la interfaz *CORBA::Object*). Por eso no se ha definido especialmente una operación con este fin en la interfaz de fachada de base.

8.2.4 Operación *packagesGet()*

La operación *packagesGet()* retorna la lista de los lotes condicionales soportados por el objeto gestionado denominado. Los objetos gestionados accesibles desde la misma fachada pueden soportar lotes condicionales diferentes. La Recomendación UIT-T X.780 define la noción de lotes condicionales, cada uno con un nombre de cadena. *StringSetType* es una definición de tipo para una secuencia de cadenas.

8.2.5 Operación *creationSourceGet()*

La operación *creationSourceGet()* retorna un valor que indica el sistema que provocó la creación del objeto gestionado denominado. *SourceIndicatorType* es un tipo enumerado que tiene tres valores: *resourceOperation*, *managementOperation*, y *unknown*. Indica si se trata de un objeto creado de forma autónoma por el recurso, en respuesta a una operación de gestión, o se desconoce el motivo para la creación del objeto.

8.2.6 Operación *deletePolicyGet()*

La operación *deletePolicyGet()* retorna la política de supresión para el objeto gestionado denominado. Es un valor enumerado que indica si el objeto no es suprimible, si sólo es suprimible cuando no contiene objetos, o si todos los objetos contenidos serán suprimidos cuando se suprima el objeto en cuestión. (La supresión de un objeto pero sin suprimir los objetos que éste contiene no está permitida.) Esta política se establece cuando la fábrica del objeto lo crea en base a la información de vinculación del nombre identificada en la operación de creación.

8.2.7 Operación *attributesGet()*

Se utiliza el método *attributesGet()* para retornar todos los valores de los atributos de un objeto, o cualquier subconjunto de estos valores en una operación. Para cada interfaz de objeto gestionado o de fachada en un modelo de información se definirá un *valuetype* CORBA que contendrá datos para cada uno de los atributos legibles en esa interfaz. (Atributos legibles son los que se utilizan con una operación *<attribute name>Get()*.) Este método puede utilizarse para obtener este tipo de valor para cualquier objeto gestionado. Estos tipos de valor se definirán según la jerarquía de herencia de las interfaces de objeto gestionado (con la salvedad de que los tipos de valor no pueden soportar varias herencias), y finalmente cada uno se derivará del *ManagedObjectValueType* definido en UIT-T X.780. El objeto gestionado retornará la subclase definida para su interfaz en respuesta a este método. Así, invocando la operación *attributesGet()* sobre cualquier objeto gestionado, un cliente obtiene en retorno una referencia a un *ManagedObjectValueType*, que podrá entonces adaptar (*cast*) al tipo de valor definido para la interfaz en la que se invocó esta operación.

La cuestión se complica por el hecho de que un cliente puede no desear recuperar todos los valores de atributos de un ejemplar, y un ejemplar puede que no soporte todos los atributos que están en los lotes condicionales. (Los tipos de valor incluyen atributos en lotes condicionales.) Esto se consigue mediante el empleo del parámetro *attributeNames* in/out. En la invocación, el cliente puede dar una lista de los nombres de atributos que le interesan, o una lista vacía para indicar que se deben retornar todos los atributos soportados. El objeto gestionado no tendrá en cuenta los nombres de esta lista que no sean nombres de atributo válidos. En su respuesta, el objeto retornará la lista efectiva de atributos para los que se han dado valores. Téngase en cuenta que esta lista puede no concordar con la lista presentada. El objeto siempre retornará una lista exacta, incluso cuando la lista presentada estuviera vacía o contuviera nombres no válidos. Si todos los nombres de la lista presentada son no válidos, el objeto debe retornar una lista nula y un tipo de valor vacío.

Como la estructura del tipo de valor está predefinida, el objeto deberá insertar algunos valores para los atributos no solicitados o no soportados. Esencialmente, el objeto puede retornar cualquier valor para estos atributos, pero, por razones de eficiencia, los valores deberán ser lo más cortos posible. Por tanto, se deben retornar valores nulos para toda clase de cadenas, referencias y listas. Se puede retornar cualquier valor para enteros y tipos enumerados. El cliente considerará no válido todo valor para un atributo no denominado en la lista retornada por el objeto.

8.2.8 Operación *attributesBulkGet()*

El método *attributesBulkGet()* se utiliza para retornar múltiples atributos de múltiples objetos gestionados del mismo tipo que son accesibles a través de la fachada. El sistema de gestión proporciona una lista de nombres de atributo y una lista de nombres de objeto gestionado de las cuales se recuperan esos atributos. La lista de nombres de atributo recibe aquí el mismo tratamiento que en la operación *attributesGet* descrita anteriormente. Más adelante se describe el tratamiento de la lista de nombres de objeto gestionado.

8.2.8.1 Determinación de los objetos a partir de los cuales se recuperan atributos

Si la lista está vacía, se solicitan implícitamente todos los objetos accesibles a través de la fachada.

Si uno de los elementos de esta lista no tiene el componente final con un valor *ID* de "Object" o <empty>, ese elemento hace las veces de un nombre parcial, que solicita implícitamente el conjunto de objetos gestionados (accesibles a través de la fachada) cuyos nombres empiecen por ese nombre parcial. Tal nombre parcial se crearía retirando uno o más componentes de nombre a partir del final de un nombre de objeto gestionado.

Por ejemplo, en el caso de un elemento gestionado cuyo nombre tiene un *ID* de "me1", utilizando un valor de *kind* "ME", para una raíz local definida para acme telecom:

```
acme\.com/me1.ME/.facadeID1
```

Obsérvese que en el componente de nombre final (después del último '/'), la cadena *ID* (que precede a '.') está vacía. Por tanto, no es un nombre parcial. Pasando al caso de un objeto Equipment "eq1" denominado bajo me1 utilizando "EQ" como kind:

```
acme\.com/me1.ME/eq1.EQ/.facadeID2
```

Por último, supóngase que hay objetos gestionados Equipment Holder (tenedor de equipo) denominados bajo este objeto de equipo (utilizando kind "EH"), como se muestra en la figura 4, accesibles todos a través de una fachada de tenedor de equipo con el ID "facadeID3".

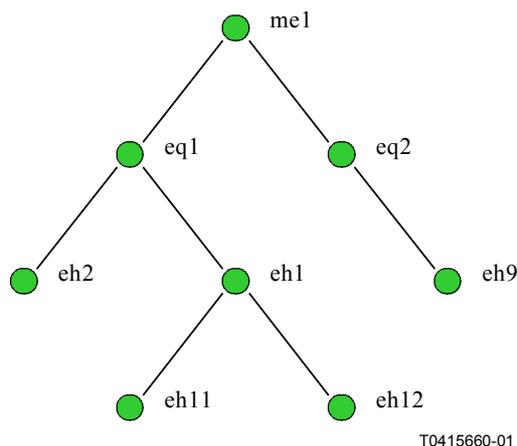


Figura 4/X.780.1 – Ejemplo de árbol de denominación

Si el nombre parcial "acme\.com/me1.ME/eq1.EQ" se utiliza en la lista de parámetros de nombres de objetos gestionados de la operación *attributesBulkGet* sobre la fachada Equipment Holder, se retornarán los valores de atributo de los siguientes objetos tenedor de equipo:

```
acme.com/me1.ME/eq1.EQ/eh1.EH/.facadeID3
```

```
acme.com/me1.ME/eq1.EQ/eh2.EH/.facadeID3
```

```
acme.com/me1.ME/eq1.EQ/eh1.EH/eh11.EH/.facadeID3
```

```
acme.com/me1.ME/eq1.EQ/eh1.EH/eh12.EH/.facadeID3
```

La fachada puede reconocer que se trata de un nombre parcial porque el valor *ID* en el componente de nombre final no es "Object" ni <empty>.

Si bien no se ha demostrado en este ejemplo, el objeto en base del árbol identificado por el nombre parcial está incluido en el alcance de la operación.

8.2.8.2 Retornar los resultados

Los datos se retornan en tipos de valores de objetos gestionados sumamente tipificados, uno de cada objeto gestionado denominado. Si la fachada no da acceso para un nombre de objeto gestionado proporcionado por el cliente, no se retorna ningún tipo de valor para ese objeto. Dado que un sistema gestionado puede proporcionar múltiples interfaces de fachada del mismo tipo, es posible que el cliente tenga que invocar esta operación en múltiples interfaces para recuperar valores de todos los objetos gestionados de un mismo tipo en un sistema.

La fachada retornará el nombre en cada tipo de valor de objeto gestionado, aunque el cliente no haya pedido que se retornen valores para el atributo "name". Se hace así para que el cliente sepa qué valores se aplican a qué objeto gestionado.

Junto con el tipo de valor de cada objeto gestionado retornado va una lista de los nombres de los atributos en ese tipo de valor que tienen valores válidos. Es posible que esta lista no concuerde con la lista de atributos solicitados, pues el ejemplar puede no soportar todos los atributos solicitados. Si el ejemplar no soporta ninguno de los atributos solicitados la fachada retornará un tipo de valor de objeto gestionado para esa instancia pero sólo el atributo de nombre contendrá un valor válido.

Dado que el volumen de datos retornados puede ser muy grande, se emplea el esquema de diseño de iterador. El cliente especifica el número máximo de tipos de valor que habrán de ser retornados. El resto se retornará en un iterador. Si se necesita un iterador, el valor de retorno será verdadero. Si no se necesita este valor será falso y la referencia de iterador será nula.

8.2.9 Operación *destroy()*

La operación *destroy()*, la última en la fachada de base, se utiliza para liberar todo recurso asociado con el objeto gestionado y suprimir este objeto. Se formula la excepción *DeleteError* si el objeto tiene una política de supresión de *NotDeletable* (no suprimible). La excepción *DeleteError* también es un medio extensible de informar problemas mediante la destrucción de objetos que dependen del modelo. Por ejemplo, tratar de suprimir un objeto punto de terminación de camino antes de que se haya suprimido el camino podría dar origen a un *DeleteError*. Sin embargo, la Recomendación UIT-T Q.816 define un servicio denominado "servicio de terminación" para implementar la lógica necesaria para aplicar las políticas de supresión y mantener la integridad del árbol de denominación. De hecho, la operación destruir está prevista para ser utilizada por este servicio, y no debe ser invocada directamente por un sistema de gestión. Véase la explicación detallada del servicio de terminación en UIT-T Q.816.

(R) FACHADA-5 – Las interfaces de fachada definidas para una interfaz CORBA de granularidad gruesa heredarán (directa o indirectamente), de la interfaz *ManagedObject_F* descrita anteriormente y definida en el CORBA IDL del anexo A. Las capacidades descritas anteriormente deberán estar soportadas.

8.3 Interfaz iteradora *AttributesBulkGet*

Como se indicó en 8.2.8, puede ser necesario un iterador si hay que retornar muchos resultados en respuesta a una operación *attributesBulkGet*. El esquema de diseño de iterador es un conocido esquema de diseño CORBA. Cuando hay que retornar grandes cantidades de datos en respuesta a una operación se retorna en su lugar una referencia a una interfaz iteradora. El cliente podrá entonces interrogar el iterador para recuperar los resultados en segmentos manejables.

La descripción IDL del iterador "attributes get result" se indica a continuación.

```
interface AttributesGetResultIterator {  
  
    boolean getNext(in unsigned short howMany,  
                   out AttributesGetResultSet results)  
        raises (ApplicationError);  
};
```

```
void destroy();  
  
}; // end of interface AttributesGetResultIterator
```

(R) FACHADA-6 – El sistema gestionado ejemplificará un iterador con una interfaz que corresponda a la descripción de la definición de *AttributesGetResultIterator* en el IDL del anexo A, cuando el número de resultados que habrá de retornarse en respuesta a una operación *attributesBulkGet* es superior al número solicitado por el cliente.

(R) FACHADA-7 – Cada vez que un cliente invoca la operación *getNext* en el iterador, éste retorna un conjunto de resultados. El iterador registra el número de resultados que ya han sido recuperados por cliente, y retorna todos los resultados una sola vez. Los resultados retornados inicialmente en respuesta a la operación *attributesBulkGet* no serán retornados nuevamente por el iterador. El iterador retornará, como máximo, el número de nombres indicado por el parámetro *howMany*, en respuesta a la operación *getNext*. El iterador puede retornar menos datos que el tamaño de lote solicitado, compensando la ventaja de retornar un lote grande de resultados con la posible necesidad de bloquear hasta que haya más resultados disponibles. Si hay más resultados por retornar (además de los que se están retornando en ese momento), el valor de retorno de la operación *getNext* será verdadero, si no los hay, este valor será falso. El iterador no retornará un conjunto de resultados vacío a menos que *howMany* haya sido fijado a cero, o cuando ya no haya más resultados por retornar, pues en tal situación el cliente tendría que interrogar el iterador.

(R) FACHADA-8 – El sistema gestionado controlará el ciclo de vida del iterador. Ahora bien, existe una operación destruir, que permite al gestor detener la recuperación de los resultados antes de llegar al último. Cuando se invoca la operación destruir, el iterador libera los recursos que esté utilizando y se suprime él mismo. El iterador también puede ser destruido por el sistema gestionado si no se ha utilizado durante un periodo de tiempo irrazonablemente largo.

8.4 Ejemplificación de fábricas

Las fábricas son interfaces de objeto persistentes que se utilizan para ejemplificar otros objetos. Estas fábricas se utilizan según un conocido esquema de diseño CORBA. En las interfaces se utilizan fábricas para proporcionar a un sistema gestor un método para crear nuevos ejemplares de objetos gestionados. Si bien no son rigurosamente necesarias en las interfaces de granularidad gruesa, porque la fachada podría desempeñar el papel de la fábrica, se utilizarán distintas interfaces de fábrica en las interfaces de granularidad gruesa. Una ventaja de esto es hacer más compatibles los tratamientos de las interfaces de granularidad gruesa y las de granularidad fina. Otra ventaja es que impiden que las subclases hereden operaciones de creación de objetos de superclase. Este problema no se presenta cuando hay distintas fábricas, porque las interfaces de fábrica no siguen la jerarquía de herencia de los objetos gestionados. Las interfaces de fábrica utilizadas para crear objetos gestionados en interfaces de granularidad gruesa son idénticas a las definidas para las interfaces de granularidad fina. Véase la cláusula 9.

(R) FACHADA-6 – Un sistema gestionado proporcionará, como mínimo, una interfaz de fábrica para cada clase de objetos gestionados que se puedan ejemplificar en él. No es necesario proporcionar interfaces de fábrica para clases de objetos gestionados que no se puedan ejemplificar en el sistema. Las interfaces de fábrica son registradas mediante el servicio buscador de fábrica, definido en UIT-T Q.816. Las fábricas en interfaces de granularidad gruesa pueden retornar una referencia nula en respuesta a una operación de creación, en vez de una referencia a un objeto de nueva creación. El nombre de objeto gestionado retornado por la fábrica en respuesta a la operación de creación indicará la fachada que se puede utilizar para acceder al nuevo objeto, según las reglas de denominación de objetos gestionados definidas en UIT-T Q.816.1.

9 Directrices para el modelado CORBA de granularidad gruesa

En esta cláusula se dan las reglas para definir interfaces IDL de granularidad gruesa. Para crear una interfaz de granularidad gruesa, se comienza por definir una interfaz de granularidad fina según las directrices de UIT-T X.780. La Recomendación UIT-T X.780 trata la creación de interfaces IDL nuevas y la traducción de una interfaz GDMO en una interfaz IDL. Una vez definida la interfaz de granularidad fina, se desarrollarán interfaces de fachada para cada una de las interfaces de objetos gestionados, como se indica en la siguiente cláusula 10. Se mantendrán las especificaciones de la interfaz de objeto gestionado de granularidad fina. Todas las otras construcciones definidas para las interfaces de granularidad fina, incluidos tipos de datos, tipos de valores, excepciones, notificaciones y fábricas, son reutilizadas sin modificación en la interfaz de granularidad gruesa.

10 Directrices para la traducción de modelos de granularidad fina en modelos de granularidad gruesa

A continuación se explica la forma de crear un modelo IDL de granularidad gruesa a partir de un modelo IDL de granularidad fina.

- 1) Se creará una interfaz de fachada para cada interfaz de objeto gestionado. Una interfaz de objeto gestionado se deriva directa o indirectamente de la interfaz *ManagedObject*.
- 2) Las interfaces de fachada se crearán dentro del mismo módulo IDL que las interfaces de objetos de granularidad fina. Esto exime al modelador de la obligación de incluir definiciones de tipo para todos los tipos definidos en el modelo de granularidad fina.

Las interfaces de fachada se pueden crear en un fichero aparte, o se pueden incluir en una nueva versión del fichero para el modelo de granularidad fina. En el primer caso, será necesario incluir, para compilación, el fichero que contiene el modelo de granularidad fina.

Las normas OMG permiten dividir un módulo en varios ficheros. Esencialmente, un módulo IDL define un espacio de nombre. En un módulo, todos los nombres deben ser únicos. Así, por ejemplo, un módulo no puede contener dos interfaces denominadas "ManagedObject_F". Ahora bien, dos módulos diferentes pueden contener nombres idénticos y unos módulos pueden estar contenidos en otros módulos. La regla de nombre único se sigue aplicando cuando un módulo se divide en varios ficheros. No se permiten nombres duplicados en el mismo módulo, aunque estén en ficheros distintos.

El IDL del anexo A está definido dentro de un solo módulo llamado "itut_x780", que es el mismo módulo utilizado en UIT-T X.780. Así pues, este módulo está dividido en varios ficheros. Una consecuencia de esto es que ninguno de los nombres utilizados en el IDL de UIT-T X.780 se puede reutilizar en el IDL de esta Recomendación. No obstante, tiene la ventaja de que las construcciones IDL definidas en UIT-T X.780 se pueden reutilizar en la presente Recomendación, simplemente incluyendo el fichero con una directiva de precompilación. No serán necesarias definiciones de tipos ni nombres confinados a un alcance, los que sí habría que incluir si el IDL estuviera en módulos distintos. Obsérvese que el cumplimiento y la conformidad siguen estando basados en documentos, no en módulos IDL. Por tanto, un sistema puede ser conforme con UIT-T X.780 y no ser conforme con la presente Recomendación.

- 3) La interfaz de fachada tendrá el mismo nombre de la interfaz de granularidad fina a partir de la cual fue creada, seguido de "_F" (un carácter de subrayado y la letra F mayúscula) . De esta forma, la interfaz de fachada creada para el objeto gestionado "Equipment" se llamará "Equipment_F".

- 4) Si la interfaz de objeto de granularidad fina hereda directamente de la interfaz *ManagedObject*, la interfaz de fachada creada para ella heredar  de la interfaz *ManagedObject_F*. En cambio, si la interfaz de objeto de granularidad fina hereda de una subclase de la interfaz *ManagedObject*, la interfaz de fachada creada para ella heredar  de la interfaz de fachada creada para esa subclase. Por tanto, la jerarqu  de herencia de las interfaces de fachada concuerda con la de las interfaces de objetos gestionados. Por ejemplo, suponiendo que la interfaz de objeto de granularidad fina *EquipmentHolder* hereda de la interfaz *Equipment*, la interfaz *EquipmentHolder_F* heredar  de la interfaz *Equipment_F*. Es necesario crear interfaces de fachada para todas las superclases del objeto antes de crear la fachada del objeto de granularidad fina.
- 5) Todo el contenido de la interfaz de objeto de granularidad fina se copiar  en la interfaz de fachada, con las siguientes modificaciones:
 - Se a adir  un par metro *in* de tipo *NameType* y denominado *name*, como el primer par metro, a cada operaci n. Este par metro se utilizar  para introducir el nombre del objeto gestionado de destino sobre el cual se ha de invocar la operaci n. Si uno de los par metros de la operaci n ya se hubiera denominada "name" se cambiar  el nombre de este par metro
 - Todo tipo de par metro o de retorno que utilice una referencia de objeto gestionado IOR debe traducirse a *NameType*. Un valor IOR de objeto gestionado ser  identificado como una referencia a una interfaz *ManagedObject* o a una interfaz de subclase. Est  desaconsejado utilizar tales tipos de valores en las interfaces de objeto de granularidad fina, por lo que normalmente no habr  ninguno, o muy pocos.
 - Para una descripci n de comportamiento en una especificaci n de interfaz de granularidad gruesa, se admite hacer referencia a una descripci n de comportamiento de la construcci n equivalente en una interfaz de granularidad fina, m s bien que duplicarla.
- 6) Las directrices para el modelado de granularidad fina actualmente proh ben la utilizaci n de atributos OMG IDL para modelar atributos de objetos gestionados. Esto se debe a que el OMG IDL no permite formular excepciones definidas por el usuario en operaciones de acceso a atributos. Por tanto, en lugar de esto, los atributos de objetos gestionados se modelan de manera que se utilicen operaciones distintas para obtener o fijar el valor del atributo, o para a adir valores al atributo o suprimir valores de un atributo. Ahora bien, es de esperar que, en el futuro, el OMG permitir  excepciones definidas por el usuario en las operaciones de acceso a atributos. De ser as , y si una interfaz de objeto de granularidad fina utiliza atributos IDL, entonces cuando se vaya a crear una fachada para este objeto, los atributos se traducir n en operaciones IDL distintas y el par metro *name* se a adir  como el primer par metro de esas operaciones.
- 7) El resto del IDL de la interfaz de granularidad fina como los tipos de datos y tipos de valores, se utiliza en la interfaz de granularidad gruesa sin modificaci n.

11 Cumplimiento y conformidad con respecto al IDL de granularidad gruesa

En esta cl usula se definen los criterios que deben ser satisfechos por otros documentos relativos a normas en los que se reivindica el cumplimiento de estas directrices, y las funciones que deben ser implementadas por los sistemas conformes con esta Recomendaci n.

11.1 Cumplimiento por parte de documentos relativos a normas

Toda especificaci n que pretenda cumplir estas directrices deber :

- 1) Satisfacer todos los requisitos de cumplimiento establecidos para los documentos relativos a normas en UIT-T X.780.

- 2) Respetar las reglas de correspondencia del IDL de granularidad fina al IDL de granularidad gruesa definidas en la cláusula 10.

11.2 Conformidad de sistema

Una implementación que pretenda ser conforme con esta Recomendación deberá:

- 1) Satisfacer todos los requisitos de ejemplificación de fachadas, fábricas e iteradores especificados en la cláusula 8.
- 2) Implementar una interfaz IDL que cumpla las directrices de esta Recomendación. Véase 11.1.

11.3 Directrices para los enunciados de conformidad

Los usuarios de estas directrices deben tener cuidado cuando formulen enunciados de conformidad. Dado que se utilizan módulos IDL como espacios de nombre, dichos módulos pueden ser divididos en varios ficheros, pues así lo permiten las reglas del OMG IDL. Por tanto, en caso de extensión de un módulo, no se cambia el nombre sino que, simplemente, se añade un nuevo fichero IDL. Por tanto, la simple indicación del nombre de un módulo en un enunciado de conformidad no será suficiente para identificar un conjunto de interfaces IDL. El enunciado de conformidad deberá identificar un documento y el año de publicación para garantizar que se identifica la versión correcta de IDL.

ANEXO A

IDL de modulado de granularidad gruesa

```
/* Este código IDL está destinado a ser almacenado en un fichero denominado
"itut_x780_1.idl" situado en el camino de búsqueda utilizado por compiladores IDL en el
sistema que usted emplea. */

#ifndef ITUT_X780_1_IDL
#define ITUT_X780_1_IDL

#include <itut_x780.idl>

#pragma prefix "itu.int"

module itut_x780 {

// IMPORTED TYPES
// DATA TYPES

/** Esta estructura contiene los resultados de recuperar un conjunto de
valores de atributo de un objeto gestionado individual. Los valores de
atributo son colocados en el miembro atributos sumamente tipificado. Puesto
que no todos los valores en los atributos pueden haber sido solicitados o
soportados, el miembro nombre de atributo contiene la lista de nombres de
atributo para los cuales el miembro atributos contiene valores válidos. Los
restantes son no válidos. */

struct AttributesGetResultType {
    ManagedObjectValueType attributes;
    StringSetType attributeNames; };

typedef sequence <AttributesGetResultType> AttributesGetResultSetType;

interface AttributesGetResultIterator;
```

// ATTRIBUTES GET RESULT ITERATOR INTERFACE

/** La interfaz Attributes Get Result Iterator se utiliza para recuperar los resultados de una operación attributesBulkGet que utiliza el esquema de diseño de iterador. */

```
interface AttributesGetResultIterator {

    /** Este método se utiliza para recuperar los siguientes "howMany"
    resultados en el conjunto de resultados.
    @param howMany    The maximum number of items to be returned in
                     the results. Fewer may be returned if that is all
                     that is left, or to balance delay with efficiency.
    @param results    The next batch of results.
    @return           True if there are more results after those being
                     returned. If the return value is true the results
                     set should not be empty, as this forces the client
                     to poll for results.
                     Instead the call should block.
    */

    boolean getNext(in unsigned short howMany,
                   out AttributesGetResultSetType results)
                   raises (ApplicationError);

    /** Este método se utiliza para destruir el iterador y liberar sus
    recursos. De todas formas el iterador es destruido automáticamente
    después de que se hayan retornado los últimos resultados, y puede ser
    destruido si no se utiliza durante un periodo de tiempo
    irrazonablemente largo. */

    void destroy();

}; // end of interface AttributesGetResultIterator
```

// MANAGED OBJECT FACADE

/** La fachada Managed Object está destinada a ser la interfaz de base de la que hereden todas las otras fachadas de objetos gestionados. Es un lugar central para especificar las funciones básicas que cabe esperar serán soportadas por todas las fachadas de objetos gestionados. */

```
interface ManagedObject_F {

    /** Este método retorna el nombre confinado al alcance de la clase de
    objeto gestionado más específica (por ejemplo
    "itut_x780::EquipmentR1").
    *NOTA* Esta operación retorna el nombre de clase del objeto, no de la
    fachada. Éste es el nombre de la interfaz de fachada sin la "_F" final.
    También es el mismo nombre que va en el parámetro objectClass de
    notificaciones, incluso en interfaces que soportan fachadas.
    @param name      The name of the managed object instance on which the
                     operation is to be invoked.
    @return          The interface name of the managed object.
    */

    ObjectClassType objectClassGet(in NameType name)
    raises (ApplicationError);

    /** Este método retorna una lista de todos los lotes condicionales
    soportados por este ejemplar.
    @param name      The name of the managed object instance on which the
                     operation is to be invoked.
    @return          The list of package names supported by the managed object
    */
}
```

```

StringSetType packagesGet (in NameType name)
    raises (ApplicationError);

/** Este método retorna una indicación de la forma de creación del
objeto.
@param name      The name of the managed object instance on which the
operation is to be invoked.
@return         An indication of whether the named managed object was
created autonomously or by a managing system
*/

```

```

SourceIndicatorType creationSourceGet(in NameType name)
    raises (ApplicationError);

```

```

/** Este método retorna un valor que indica si el objeto se puede
suprimir y, en su caso, si todos los objetos contenidos se suprimen
automáticamente.
@param name      The name of the managed object instance on which the
operation is to be invoked.
@return         The delete policy of the named managed object
*/

```

```

DeletePolicyType deletePolicyGet (in NameType name)
    raises (ApplicationError);

```

```

/** Este método puede utilizarse para obtener de forma general todos
los atributos soportados por un ejemplar. Se espera que cada interfaz
forme una subclase del tipo de valor Managed Object y añada los otros
atributos soportados por esa interfaz. El objeto gestionado debe
retornar un objeto de valor de ese tipo. El cliente debe entonces
precisar la referencia para acceder a todos los atributos. <p>

```

El cliente también puede presentar una lista de nombres que indique los atributos que desea recibir. Estos nombres deben concordar con los nombres de miembro en el objeto de valor. Para los miembros que no estén en la lista, y para los miembros que formen parte de lotes que no estén soportados, el servidor puede retornar cualquier valor, pero éste debe ser lo más corto posible. El servidor también retorna la lista de atributos, que puede ser más corta por la exclusión de atributos en lotes no soportados. El cliente debe considerar que el valor de un miembro que no esté en la lista retornada no tiene ningún significado.

Una lista de nombres de atributos nula indica que todos los atributos soportados habrán de ser retornados. El servidor retornará la lista real.

```

@param name      The name of the managed object instance on
which the operation is to be invoked.
@param attributeNames  A list of names of attributes to be retrieved.
@return         The value type containing the attributes.
*/

```

```

ManagedObjectValueType attributesGet (
    in      NameType name,
    inout   StringSetType attributeNames)
    raises (ApplicationError);

```

```

/** Este método se utiliza para retornar múltiples atributos de
múltiples objetos gestionados del mismo tipo. El cliente da una lista
de nombres de atributos y una lista de nombres de objetos gestionados
de las que se han de recuperar los atributos. <p>

```

Los datos son retornados en tipos de valor de objetos gestionados sumamente tipificados, uno de cada objeto gestionado denominado. Si la fachada no da acceso para un nombre de objeto gestionado proporcionado por el cliente, no se retorna ningún tipo de valor para ese objeto. Dado que un sistema gestionado puede proporcionar múltiples interfaces de fachada del mismo tipo, es posible que el cliente tenga que invocar esta operación en múltiples interfaces para recuperar valores de todos los objetos gestionados de un determinado tipo en un sistema. <p>

La fachada retornará el nombre en todos los tipos de valor objetos gestionados, incluso si el cliente no pide que se retornen valores para el atributo 'name'. De otra forma, el cliente no sabrá qué valores aplicar a qué ejemplar de objeto gestionado. <p>

Junto con el valor de cada objeto gestionado retornado va una lista de los nombres de los atributos de ese tipo de valor que tienen valores válidos. Es posible que esta lista no concuerde con la lista de atributos pedidos, pues el ejemplar puede no soportar todos los atributos pedidos. El valor "name" siempre estará en la lista retornada. Si el ejemplar no soporta ninguno de los atributos pedidos, la fachada retornará un tipo de valor de objeto gestionado para ese ejemplar, donde sólo el atributo name contendrá un valor válido. <p>

Dado que puede ser retornada una cantidad posiblemente grande de datos se utiliza el esquema de diseño de iterador. El cliente especifica el número máximo de tipos de valor que habrán de ser retornados. El resto será retornado en un iterador. Si se utiliza un iterador, el valor de retorno será verdadero. De lo contrario, será falso y la referencia de iterador será nula.

```
@param names           The names of the managed objects from which to
                        retrieve the attribute values.
@param attributeNames  The names of the attributes to retrieve.
@param howMany         The maximum number of value types to return in
                        the attributes parameter.
@param attributes      The first batch of results.
@param iterator        A reference to an iterator, if needed.
                        Otherwise, null.
@return                True if an iterator is being returned,
                        otherwise false.
```

```
*/
```

```
boolean attributesBulkGet (
    in      NameSetType      names,
    in      StringSetType    attributeNames,
    in      unsigned short   howMany,
    out     AttributesGetResultSetType  attributes,
    out     AttributesGetResultIterator  iterator)
    raises (ApplicationError);
```

```
/** Este método destruye el objeto. Se utiliza, simplemente, para
liberar todo recurso asociado con el objeto gestionado. No efectúa
comprobaciones para determinar si hay objetos contenidos, ni suprime
vinculaciones de nombres del árbol de denominación. <p>
```

Esta operación tiene por finalidad permitir que servicios de soporte destruyan el objeto gestionado. <p>

NOTA - La invocación directa de esta operación por un sistema de gestión podría corromper el árbol de denominación y sólo se recomienda en circunstancias extraordinarias. Los clientes que deseen suprimir un objeto deben, en lugar de esto, utilizar el servicio terminador.

```
@param name          The name of the managed object instance on
                        which the operation is to be invoked.
```

```
*/
```

```

        void destroy(in NameType name)
            raises      (ApplicationError,
                       DeleteError);

}; // end of ManagedObject_F interface

// ApplicationErrorConst Module

/** Este módulo contiene las constantes definidas para el código de error
contenido en las estructuras Application Error Info retornadas con excepciones
Application Error.
*/

module ApplicationErrorConst {

    /** Este código de excepción de error de aplicación indica que no se pudo
    encontrar un objeto de una operación. */

    const short objectNotFound = 4;

}; // end of module ApplicationErrorConst

}; // end of module itut_x780

#endif // end of #ifndef ITUT_X780_1_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 multimedios
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedios
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