UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T Q.816.1

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

(08/2001)

SERIE Q: CONMUTACIÓN Y SEÑALIZACIÓN

Interfaz Q3

Servicios de la RGT basados en arquitectura de intermediario de petición de objeto común: Extensiones para el soporte de interfaces de granularidad gruesa

Recomendación UIT-T Q.816.1

RECOMENDACIONES UIT-T DE LA SERIE Q

CONMUTACIÓN Y SEÑALIZACIÓN

SEÑALIZACIÓN EN EL SERVICIO MANUAL INTERNACIONAL	Q.1–Q.3
EXPLOTACIÓN INTERNACIONAL SEMIAUTOMÁTICA Y AUTOMÁTICA	Q.4–Q.59
FUNCIONES Y FLUJOS DE INFORMACIÓN PARA SERVICIOS DE LA RDSI	Q.60–Q.99
CLÁUSULAS APLICABLES A TODOS LOS SISTEMAS NORMALIZADOS DEL UIT-T	Q.100–Q.119
ESPECIFICACIONES DE LOS SISTEMAS DE SEÑALIZACIÓN N.° 4 Y N.° 5	Q.120–Q.249
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.° 6	Q.250–Q.309
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN R1	Q.310–Q.399
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN R2	Q.400–Q.499
CENTRALES DIGITALES	Q.500–Q.599
INTERFUNCIONAMIENTO DE LOS SISTEMAS DE SEÑALIZACIÓN	Q.600–Q.699
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.° 7	Q.700–Q.799
Generalidades	Q.700
Parte transferencia de mensajes	Q.701–Q.709
Parte control de la conexión de señalización	Q.711–Q.719
Parte usuario de telefonía	Q.720–Q.729
Servicios suplementarios de la RDSI	Q.730–Q.739
Parte usuario de datos	Q.740–Q.749
Gestión del sistema de señalización N.° 7	Q.750–Q.759
Parte usuario de la RDSI	Q.760–Q.769
Parte aplicación de capacidades de transacción	Q.770–Q.779
Especificaciones de las pruebas	Q.780–Q.799
INTERFAZ Q3	**Q.800–Q.849**
SISTEMA DE SEÑALIZACIÓN DIGITAL DE ABONADO N.° 1	Q.850–Q.999
RED MÓVIL TERRESTRE PÚBLICA	Q.1000–Q.1099
INTERFUNCIONAMIENTO CON SISTEMAS MÓVILES POR SATÉLITE	Q.1100–Q.1199
RED INTELIGENTE	Q.1200–Q.1699
REQUISITOS Y PROTOCOLOS DE SEÑALIZACIÓN PARA IMT-2000	Q.1700–Q.1799
ESPECIFICACIONES DE LA SEÑALIZACIÓN RELACIONADA CON EL CONTROL DE LLAMADA INDEPENDIENTE DEL PORTADOR	Q.1900–Q.1999
RED DIGITAL DE SERVICIOS INTEGRADOS DE BANDA ANCHA (RDSI-BA)	Q.2000–Q.2999

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

Recomendación UIT-T Q.816.1

Servicios de la RGT basados en arquitectura de intermediario de petición de objeto común: Extensiones para el soporte de interfaces de granularidad gruesa

Resumen

En la presente Recomendación se definen las extensiones al conjunto de servicios CORBA de la red de gestión de telecomunicaciones (RGT) necesarias para el soporte de interfaces de granularidad gruesa. Se especifica la manera en que se utilizan los servicios de objetos comunes CORBA para el soporte de interfaces de granularidad gruesa y se definen las extensiones a los servicios de soporte específicos de la RGT definidos en UIT-T Q.816. Se proporciona un módulo IDL CORBA que define las interfaces con los servicios de soporte específicos de la RGT.

Orígenes

La Recomendación UIT-T Q.816.1, 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), granularidad gruesa, interfaces de la RGT, lenguaje de definición de interfaz (IDL), objetos gestionados, procesamiento distribuido, servicios CORBA.

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.

ÍNDICE

Recomendación UIT-T Q.816.1

Servicios de la RGT basados en arquitectura de intermediario de petición de objeto común: Extensiones para el soporte de interfaces de granularidad gruesa

1 Alcance

La arquitectura de la RGT definida UIT-T M.3010 (2000) introduce conceptos a partir del procesamiento distribuido e incluye la utilización de múltiples protocolos de gestión. Las Recomendaciones UIT-T Q.816 y X.780 definen ulteriormente 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.

En esta Recomendación, junto con UIT-T X.780.1, se añaden especificaciones al marco para permitirle el soporte de un estilo de interacción entre los sistemas de gestión y los sistemas gestionados algo diferente del especificado en los documentos del marco original. Este estilo de interacción tiene ciertas ventajas, siendo la más importante que alivia al sistema de gestión de tener que recuperar una dirección de soporte lógico orientado a objeto para cada recurso gestionable al que desea acceder. En los sistemas grandes estas direcciones de soporte lógico pueden llegar a ser millones. También es algo diferente la manera en que el soporte lógico está estructurado en los sistemas gestionados, que pueden preferir algunos suministradores de sistemas gestionados.

El ámbito de aplicación de esta Recomendación es el mismo que el del marco CORBA de la RGT original. El marco y estas extensiones abarcan todas las interfaces RGT en las que se puede utilizar CORBA. Se espera, sin embargo, que no todas las capacidades y servicios aquí definidos sean necesarios en todas las interfaces RGT. Esto implica que el marco se puede utilizar para interfaces entre sistemas de gestión en todos los niveles de abstracción (entre y dentro de administraciones) así como 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 pueda utilizarse en una gama más amplia de aplicaciones. Las extensiones permiten un modo de interacción algo diferente entre los sistemas de gestión y los sistemas gestionados, lo que en muchas situaciones puede resultar más adecuado. Por consiguiente, esta Recomendación está concebida para que la utilicen los diversos grupos que especifican las interfaces de gestión de redes.

1.2 Aplicación

El planteamiento seguido en las Recomendaciones para el marco CORBA de la RGT consiste en modelar recursos de red gestionables como objetos de soporte lógico accesibles mediante CORBA. Los modelos de información escritos en el lenguaje de definición de interfaces (IDL, *interface definition language*) CORBA describen las interfaces de objetos.

CORBA proporciona transparencia con respecto a la ubicación, lo que permite la interacción entre objetos de soporte lógico independientemente de sus ubicaciones. A cada objeto se accede utilizando lo que en CORBA se denomina Referencia de objeto interoperable (IOR, *interoperable object reference*).

El marco original CORBA de la RGT modela cada recurso gestionable como un objeto CORBA independiente, cada uno con su IOR exclusiva. Este método flexible permite que cada objeto esté ubicado en cualquier lugar. Esto, sin embargo, obliga a que los sistemas de gestión dispongan de una IOR para cada objeto al que deseen acceder. Esto es una carga que muchas empresas y

administraciones del sector de las telecomunicaciones han tratado de evitar. También podría requerir que los sistemas gestionados soporten un gran número de las IOR, lo que algunos suministradores de dichos sistemas quisieran evitar. Esta Recomendación, junto con UIT-T X.780.1, define la manera en que se ha de extender el marco CORBA de la RGT para evitar la necesidad de disponer de un gran número de las IOR.

Las interfaces basadas en CORBA que utilizan el planteamiento en el que cada recurso gestionable se direcciona con una única IOR han pasado a denominarse interfaces de "granularidad fina". En cambio, aquellas en las que no se asigna una IOR a cada recurso gestionable se conocen como interfaces de "granularidad gruesa".

Dado que UIT-T X.780.1 define un planteamiento algo diferente para el modelado de recursos gestionables en interfaces de granularidad gruesa, las especificaciones del modelo de interfaz serán algo diferentes para los planteamientos de granularidad fina y de granularidad gruesa.

1.3 Estructura de la Recomendación

La presente Recomendación tiene la siguiente estructura:

Cláusula 1 Introducción, estructura y actualizaciones.

Cláusula 2 Referencias.

Cláusulas 3 y 4 Definiciones de términos y abreviaturas que se utilizan en el resto de esta Recomendación.

Cláusula 6 Consideraciones de diseño que deben tenerse en cuenta cuando se añade al marco el soporte de interfaces de granularidad gruesa.

Cláusula 7 Visión general del marco CORBA de la RGT y de los requisitos que deben satisfacer las interfaces de granularidad gruesa.

Cláusula 8 Requisitos para la utilización de los servicios de objetos comunes OMG para el soporte de las interfaces de gestión de red de granularidad gruesa.

Cláusula 9 Requisitos para la utilización de los servicios de objetos específicos de la RGT para el soporte de las interfaces de gestión de red de granularidad gruesa. El marco original definía algunos nuevos servicios que tendrán que ser ampliados para el soporte de las interfaces de granularidad gruesa. Además, se necesita un nuevo servicio.

Cláusula 10 Directrices para el cumplimiento y la conformidad.

Anexo A IDL de servicio de soporte de granularidad gruesa específico de la RGT.

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 Q.816 (2001), *Servicios de la RTG basados en arquitectura de intermediario de petición de objeto común*.

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

[3] UIT-T X.780.1 (2001), *Directrices de la RGT para la definición de interfaces de objetos gestionados arquitectura de intermediario de petición de objeto común de granularidad gruesa*.

[4] OMG Document formal/99-10-07, *The Common Object Request Broker: Architecture and Specification*, Revision 2.3.1.

[5] OMG Document formal/01-02-65, *Naming Service Specification*.

[6] OMG Document formal/00-06-20, *Notification Service Specification*, Version 1.0.

[7] OMG Document formal/00-01-04, *Telecom Log Service Specification*, Version 1.0.

[8] OMG Document formal/00-06-25, *Security Services Specification*, Version 1.5.

[9] OMG Document formal/00-06-28, *Transaction Service Specification*, Version 1.1.

[10] OMG TC Document orbos/98-05-05, *CORBA Messaging*.

[11] IETF RFC 2246 (1999), *The TLS Protocol Version 1.0*.

3 Definiciones

3.1 Términos definidos en UIT-T X.701

Los siguientes términos que se utilizan en la presente Recomendación están definidos en UIT-T X.701:

– clase de objeto gestionado;

– gestor;

– agente.

3.2 Términos definidos en UIT-T X.703

El siguiente término que se utiliza en la presente Recomendación está definido en UIT-T X.703:

– notificación.

3.3 Términos definidos en UIT-T Q.816

El siguiente término que se utiliza en la presente Recomendación está definido en UIT-T Q.816:

– canal de eventos.

4 Abreviaturas

En esta Recomendación se utilizan las siguientes siglas:

AMI Invocación de mensajería asíncrona (*asynchronous messaging invocation*)

API Interfaz de programación de aplicación (*application programming interface*)

CMIP Protocolo común de información de gestión (*common management information protocol*)

CORBA Arquitectura de intermediario de petición de objeto común (*common object request broker architecture*)

COS Servicios de objetos comunes (*common object services*)

DN Nombre distinguido (*distinguished name*)

EMS Sistema de gestión de elemento (*element management system*)

GDMO Directrices para la definición de objetos gestionados (*guidelines for the definition of managed objects*)

ID Identificador

IDL Lenguaje de definición de interfaz (*interface definition language*)

IIOP Protocolo de interoperabilidad de Internet (*Internet interoperability protocol*)

IOR Referencia de objeto interoperable (*interoperable object reference*)

MO Objeto gestionado (*managed object*)

MOO Operación de objetos múltiples (*multiple object operation*)

NE Elemento de red (*network element*)

NMS Sistema de gestión de red (*network management system*)

OAM&P Operaciones, administración, mantenimiento y aprovisionamiento (*operations, administration, maintenance and provisioning*)

OID Identificador de objeto (*object identifier*)

OMG Grupo de gestión de objetos (*object management group*)

ORB Intermediario de petición de objetos (*object request broker*)

OSI Interconexión de sistemas abiertos (*open systems interconnection*)

PDU Unidad de datos de protocolo (*protocol data unit*)

PM Gestión de la calidad de funcionamiento (*performance management*)

POA Adaptador de objeto portátil (*portable object adapter*)

QoS Calidad de servicio (*quality of service*)

RDN Nombre distinguido relativo (*relative distinguished name*)

RGT Red de gestión de las telecomunicaciones

SSL Capa de zócalos seguros (*secure socket layer*)

TII Invocación independiente del tiempo (*time-independent invocation*)

TLS Seguridad de la capa de transporte (*transport layer security*)

UIT-T Unión Internacional de Telecomunicaciones – Sector de Normalización de las Telecomunicaciones

5 Convenios

5.1 Convenios seguidos en la Recomendación

En la presente Recomendación se siguen algunos convenios para orientar al lector sobre el propósito del texto. Aunque la Recomendación en su mayor parte es normativa, los párrafos que formulan sucintamente requisitos obligatorios que ha de cumplir un sistema de gestión (sistema gestor o gestionado) empiezan con una "R" en negritas encerrada entre paréntesis, seguida de un breve nombre que indica el tema del requisito, y un número. Por ejemplo:

(R) EXAMPLE-1 Ejemplo de requisito obligatorio.

Los requisitos cuya aplicación a un sistema de gestión es facultativa son precedidos por una "O" en lugar de una "R". Por ejemplo.

(O) OPTION-1 Un ejemplo de requisito facultativo.

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

En esta Recomendación se incluyen muchos ejemplos escritos en código IDL CORBA, y en el anexo A se incluye un IDL que especifica los servicios específicos de la RGT y los tipos de datos de soporte. El IDL está escrito en el tipo de letra courier de 9 puntos:

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

5.2 Compilación del IDL

Una de las ventajas de utilizar IDL para especificar interfaces de gestión de red es que IDL se puede "compilar" en código de programación mediante herramientas que acompañan a un ORB. Esto automatiza realmente el desarrollo de una parte del código necesario para la interoperación de las aplicaciones de gestión de red. La presente Recomendación incluye un anexo que contiene código que los implementadores querrán extraer y compilar. El anexo A es normativo y los diseñadores que implementan sistemas conformes con esta Recomendación deben utilizarlo. El IDL que aparece en esta Recomendación se ha comprobado con dos compiladores para garantizar su calidad de correcto. Hay que utilizar un compilador que soporte la versión CORBA especificada en UIT-T Q.816.

Se ha dado un formato al anexo A para que sea fácil cortarlos y pegarlos en ficheros de texto simple que posteriormente se pueden compilar. A continuación se hacen sugerencias sobre la manera de hacer esto.

1) Cortar y pegar parecen funcionar mejor con la versión Microsoft® Word® de esta Recomendación. Parece que cortar y pegar con el formato de fichero Adobe® Acrobat® incluyen encabezados y pies de página, que no pueden ser compilados.
2) Todo el anexo A, desde la línea "/* This IDL code …" hastal el final, se debe almacenar en un fichero con el nombre "itut_q816_1.idl" en un directorio en el que será encontrado por el compilador IDL.
3) No es necesario suprimir los títulos incluidos en el anexo. Están encapsulados en comentarios IDL y serán ignorados por el compilador.
4) Los compiladores que realizan la conversión de IDL a HTML reconocen los comentarios que empiezan por la secuencia especial "/**". Frecuentemente, estos comentarios tienen instrucciones de formato especiales para estos compiladores. Quienes trabajan con IDL pueden querer generar HTML, pues los ficheros HTML resultantes tienen enlaces que permiten la navegación rápida a través de los ficheros.
5) El anexo ha sido formatado con tabuladores de ocho espacios y cambios de reglón fijos, lo que debe permitir a casi todos los editores de texto trabajar con este texto.

6 Consideraciones de diseño de interfaces de granularidad gruesa

En esta cláusula se enumeran varias consideraciones de diseño que deben tenerse en cuenta en el marco cuando se añade el soporte para interfaces de granularidad gruesa.

6.1 Número reducido de las IOR

Cuando se añade el soporte para interfaces de granularidad gruesa al marco, éste debe permitir que aumente el número de recursos gestionados (como por ejemplo puntos de terminación) sin que aumente el número de las IOR expuestas a través de la interfaz de gestión.

6.2 Aptitud para obtener las IOR

Una gran parte de los primeros debates sobre las ventajas de las interfaces de granularidad gruesa giraba en torno a la necesidad de reducir el número de las IOR soportadas por un sistema gestionado. Esto fue porque a lo largo de la mayor parte de los intermediarios de petición de objetos (ORB, *object request brokers*) de los años 1990 no ofrecían a los diseñadores de sistemas ninguna manera normalizada de almacenar persistentemente el estado de un objeto entre invocaciones de métodos. Por lo tanto, todos los objetos con las IOR pendientes tenían que mantenerse en memoria, lo que limitaba el número de IOR que un sistema podía soportar. No obstante, el OMG solucionó este problema con la especificación CORBA 2.2 y actualmente la mayoría de los ORB soportan la norma adaptador de objeto portátil (POA, *portable object adapter*) del OMG, que permite a los sistemas almacenar persistentemente el estado del objeto entre invocaciones de métodos. Por consiguiente, el número de objetos que un sistema gestionado puede soportar ahora está limitado esencialmente por el número de objetos que puede almacenar en su espacio de disco.

Esto, sin embargo, no anula todas las ventajas de las interfaces de granularidad gruesa. El sistema de gestión resulta ser el principal beneficiado. Cuando un sistema de gestión utiliza una interfaz CORBA de granularidad fina, para interactuar con cada recurso gestionado debe en algún momento obtener la IOR de cada uno. En un sistema que deba gestionar millones de recursos, esta cifra asciende a millones de las IOR. Los sistemas de gestión aplicarán probablemente diversas estrategias para el tratamiento de grandes cantidades de IOR. La más sencilla será únicamente resolver el nombre de un recurso gestionado hasta obtener su IOR antes de cada interacción, pero este proceso es lento y constituye una pérdida de recursos de la red de comunicaciones de datos. Otra posibilidad es obtener todos los nombres con sus IOR asociadas de una vez y almacenarlos en el sistema de gestión. Otra estrategia consiste en conservar los nombres y las IOR en una memoria cache del sistema de gestión, en la que se mantengan las IOR más recientemente utilizadas para acceder rápidamente a ellas, y se descarten aquellas que no se han utilizado durante cierto tiempo, las cuales habrán de obtenerse de nuevo del sistema gestionado cuando vuelvan a necesitarse. También podría recurrirse a otras estrategias.

Las interfaces de granularidad gruesa tienen la ventaja de que los sistemas de gestión no tienen que aplicar tales esquemas. Permiten que un sistema de gestión obtenga inicialmente y almacene sólo unas pocas IOR de un sistema gestionado. Sin embargo, para ello el sistema de gestión debe posteriormente poder determinar únicamente a partir del nombre del recurso gestionado, cuál de las IOR anteriormente obtenidas debe utilizar para interactuar con ese recurso. Es decir, dado un nombre, el sistema de gestión debe poder obtener la IOR de la interfaz para ese determinado recurso gestionado. En cambio, si el sistema de gestión tiene que consultar, utilizando el nombre, al sistema gestionado para obtener la IOR correspondiente, entonces el sistema de gestión tiene que aplicar los esquemas descritos anteriormente y se pierde una gran parte de las ventajas de añadir al marco el soporte para interfaces de granularidad gruesa.

6.3 Aplicación de servicios del marco

El soporte de interfaces de granularidad gruesa debe añadirse al marco de una manera que permita aplicar los servicios del marco existentes (por ejemplo, el servicio de operación de objeto múltiple y el servicio terminador). Esto no excluye que se puedan realizar cambios en las implementaciones de esos servicios para que soporten interfaces de granularidad gruesa.

6.4 Distinción entre los dos tipos de objetos

Los sistemas de gestión deben poder distinguir entre recursos gestionados con interfaces de granularidad fina y con interfaces de granularidad gruesa.

6.5 Denominación jerárquica

La denominación jerárquica (basada en contenencia) de recursos gestionados con interfaces de granularidad gruesa debe estar soportada.

6.6 Migración de entidades modeladas a través de los métodos relativos a la granularidad

El marco debe permitir que las implementaciones que puedan migrar accedan a un tipo particular de recurso gestionado del método de granularidad gruesa al método de granularidad fina, o viceversa.

6.7 Migración de entidades modeladas a través de tecnologías

El marco debe permitir que las implementaciones migren a diversas tecnologías de soporte lógico, como C++ o JavaBeans.

6.8 Nombres distinguidos equivalentes

El nombre distinguido de un recurso gestionado no debe depender de si al recurso se accede a través de una interfaz de granularidad fina o de granularidad gruesa. Además, debe ser posible recuperar información de la relación de contenencia de una ubicación única independientemente de si a los recursos del sistema gestionado se accede a través de interfaces de granularidad fina o de granularidad gruesa.

7 Visión general del marco y los requisitos

En la cláusula 6 se esbozaron las consideraciones de diseño que deben resolverse cuando se añade al marco el soporte para interfaces de granularidad gruesa. En esta cláusula y en el resto de esta Recomendación se proporciona una información detallada sobre la manera en que se ampliará el marco para abordar estos temas. En UIT-T X.780.1 se describe la forma de modelar las interfaces de granularidad gruesa. En primer lugar, se presenta una breve visión general del marco actual y después una visión general de las extensiones.

7.1 Visión general del marco

El marco para las interfaces RGT basadas en CORBA es un grupo de capacidades. Una pieza central del marco es un conjunto de servicios de objetos comunes OMG. El marco define la función de estos servicios en interfaces de gestión de red y define convenios para su utilización. El marco también define servicios de soporte que no se han normalizado como servicios de objetos comunes OMG, pero que se espera que serán normalizados en interfaces de gestión de red conformes con el marco.

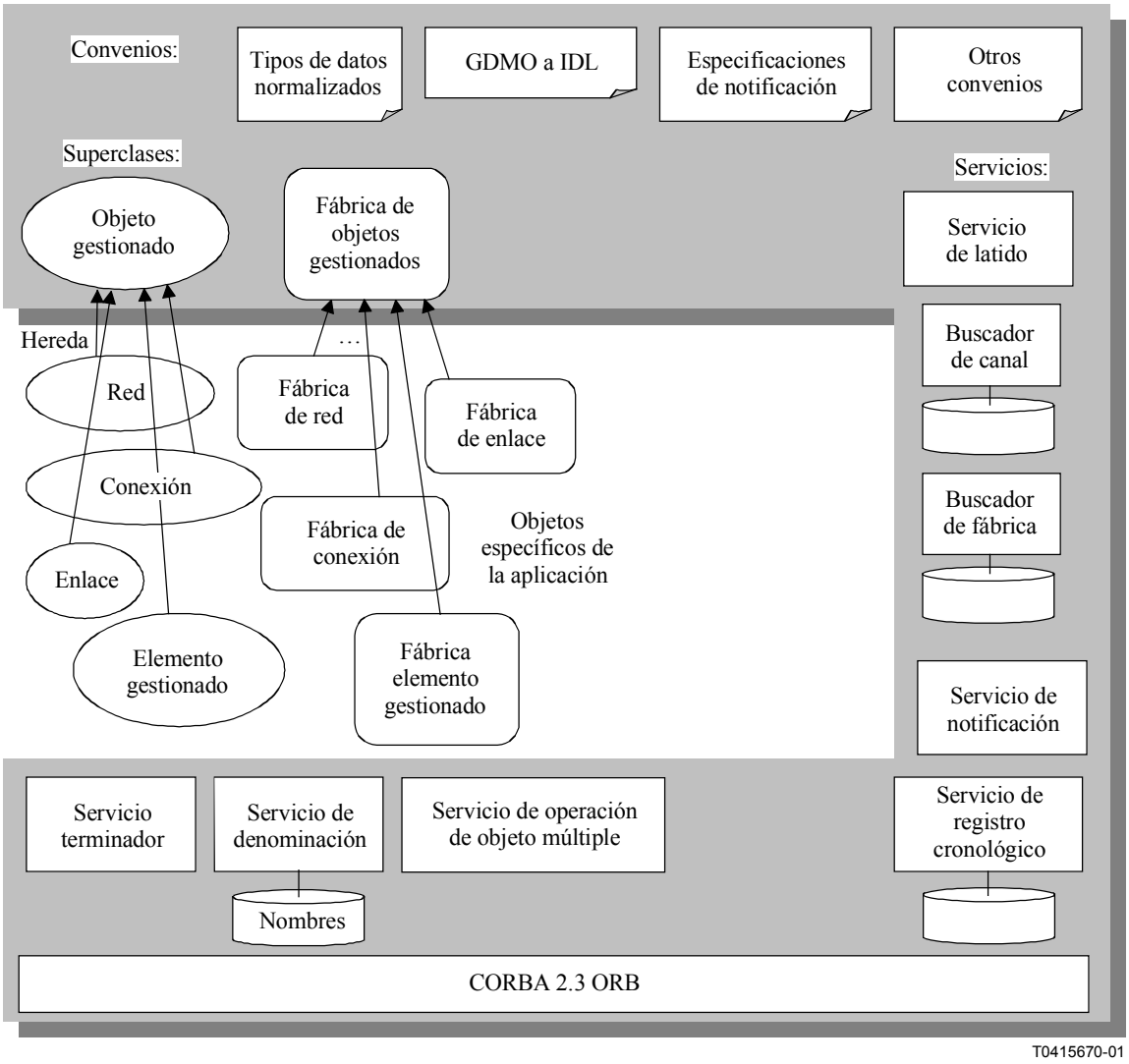

Figura 1/Q.816.1 – Visión general del marco

El marco se representa gráficamente en la figura 1 *supra*. El marco está en gris. En el centro están los objetos específicos de la aplicación que están soportados por el marco. A lo largo de la parte inferior hay una casilla que representa el CORBA ORB. Por encima hay una serie de casillas con nombres que representan los servicios que forman el marco. (Algunos también tienen iconos que ilustran las bases de datos que tendrían que mantener para ejecutar sus funciones.) Estos servicios, junto con los requisitos de versión de ORB, están definidos en UIT-T Q.816. A lo largo de la parte superior de la figura hay iconos que representan dos superclases, uno para objetos gestionados y el otro para fábricas de objetos gestionados. Cada uno de los objetos gestionados y fábricas de objetos gestionados que soporta este marco debe en último término heredar, respectivamente, de estas superclases. En la figura también aparecen iconos de páginas con el borde inferior derecho doblado que representan convenios de modelado de objeto normalizados. Estos convenios y las superclases se definen en UIT-T X.780.

7.2 Visión general de las extensiones de granularidad gruesa

En esta cláusula se presenta una visión general de las extensiones que se han de añadir al marco para el soporte de interfaces de granularidad gruesa.

7.2.1 El esquema de diseño de fachada

El cambio más importante que se ha de realizar en el marco para el soporte de interfaces de granularidad gruesa está en la manera en que se accede a los objetos gestionados. Debe ser posible aumentar el número de objetos gestionados por el sistema sin que aumente el número de IOR soportadas por el sistema. No obstante, sigue siendo deseable que el acceso a los objetos gestionados permanezca intensamente tipificado. Esto da lugar a la utilización de un esquema de diseño denominado aquí "esquema de fachada". Una fachada se puede imaginar como un frente falso, o como un portal. Utilizando este esquema de diseño, un sistema gestionado soportará un pequeño número de interfaces de fachada, por lo menos una, pero generalmente unas pocas para cada tipo de objeto gestionado en el sistema. Por lo tanto, un sistema de gestión invocará una operación sobre un objeto gestionado invocando, en realidad, la operación sobre una fachada para ese tipo de objeto gestionado en el sistema. En el esquema de diseño de fachada, los objetos gestionados no tienen que exponer una interfaz CORBA y en consecuencia puede que no tengan IOR individuales. Esto significa que un sistema gestionado que soporta el método de fachada no necesita implementar las interfaces de objetos gestionados de granularidad fina.

Es mejor imaginarse una fachada no como un objeto gestionado, sino como 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. Los objetos gestionados, sin embargo, pueden no tener interfaces CORBA y quizá no sean accesibles directamente mediante CORBA. La fachada propiamente dicha no representa un recurso de red gestionable; su finalidad es permitir la interacción con los objetos que sí representan recursos gestionables. Todos los objetos fachada los crea automáticamente el sistema gestionado y existen mientras que los objetos gestionados son accesibles a través de la fachada. En una interfaz de granularidad gruesa pueden existir múltiples fachadas para el mismo tipo de objetos gestionados, pero un objeto gestionado será accesible únicamente a través de una, y sólo una, fachada. Véase la figura 2, *infra*.

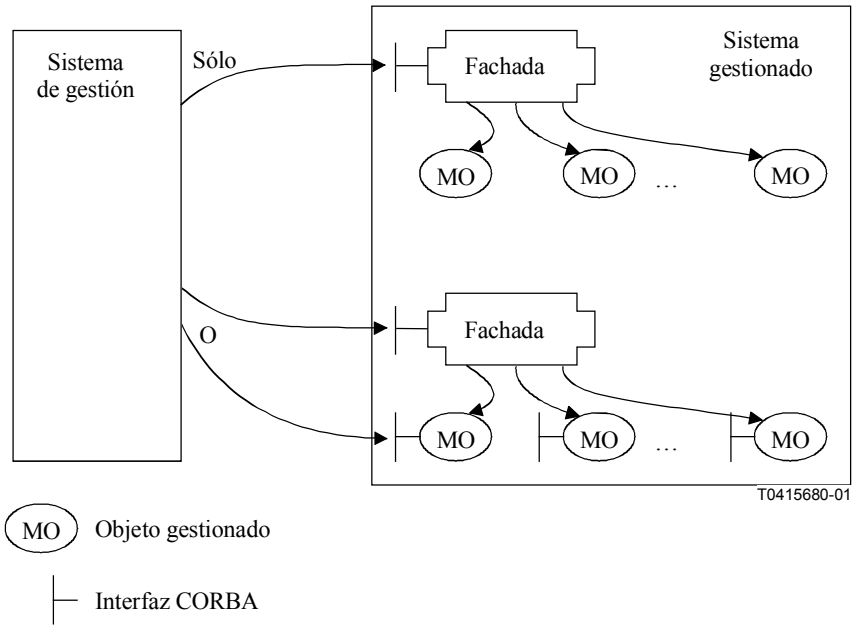

Figura 2/Q.816.1 – Papel desempeñado por la fachada

La figura muestra un sistema de gestión que accede a un sistema gestionado que soporta el método de granularidad gruesa. El sistema gestionado tiene dos interfaces de fachada que permiten al sistema de gestión acceder a dos conjuntos diferentes de objetos gestionados. Los objetos gestionados que aparecen en la parte superior de la figura solo son accesibles a través de la fachada.

Los objetos gestionados en la parte inferior de la figura también soportan interfaces CORBA directas y son accesibles a través de la fachada o directamente. El acceso CORBA directo es facultativo, pero un sistema gestionado que soporta el método de fachada debe proporcionar interfaces de fachada para cada uno de sus ejemplares de objeto gestionado.

Una fachada puede utilizar una interfaz CORBA de un objeto gestionado para invocar una operación sobre el mismo, o algún otro medio específico de la implementación. Un sistema gestionado, en realidad, ni siquiera necesita implementar objetos gestionados como objetos individuales, internamente. No obstante, al implementar una interfaz sobre la base de este marco parecerá que los objetos gestionados están implementados internamente como objetos.

Cuando se invoca una operación sobre un objeto gestionado a través de una fachada, ésta debe entonces invocar la operación sobre el objeto o entidad gestionados reales. Dado que a través de una sola fachada se accede a muchos objetos gestionados, la fachada debe saber cuál de los objetos gestionados es el verdadero objetivo de la operación. Esto se resuelve adoptando el convenio de incluir el nombre del objeto gestionado que constituye el objetivo de la operación como el primer parámetro de cada operación dirigida a un objeto gestionado.

Aunque puede que los objetos gestionados ya no tengan IOR únicas, seguirán manteniendo nombres únicos y todavía pueden considerarse como entidades individuales que representan recursos gestionables.

7.2.2 Extensión de nombre de objeto gestionado

Según se ha dicho anteriormente, los objetos gestionados a los que se accede a través de una fachada continúan teniendo un nombre aunque puede que no tengan una interfaz CORBA individual. Es importante que los sistemas de gestión puedan determinar la fachada que han de utilizar, a partir del nombre del objeto gestionado. Si no pueden, tendrán que consultar al sistema gestionado o asociar persistentemente una IOR de fachada a cada nombre de objeto gestionado. Para poder determinar la fachada de un objeto gestionado a partir únicamente de su nombre, los nombres de objetos gestionados accesibles a través de una fachada se hacen un poco más largos que los nombres de los objetos gestionados que no son accesibles a través de una fachada. En el componente de nombre final, que siempre tiene una cadena *ID* con el valor "Object" (o, según la extensión que figura en esta Recomendación, <empty>), la cadena *kind* se fija al valor de un identificador de fachada asignado a la fachada a través del cual se puede acceder al objeto. En el caso de objetos gestionados no accesibles a través de una fachada, esta cadena *kind* está vacía. En la cláusula 8.1.2 se dan más detalles sobre la forma en que se utiliza el ID de fachada 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 serán en gran parte los mismos definidos en UIT-T Q.816. Algunos, como los servicios buscador de fábrica y buscador de canal, no requieren ningún cambio en absoluto. Otros, como los servicios terminador y de operación de objetos múltiples (MOO, *multiple object operation*), no requieren ningún cambio en sus interfaces o en la manera en que los utilizan los sistemas de gestión, pero pueden requerir pequeños cambios en su implementación si acceden a objetos gestionados utilizando las interfaces fachada de objetos gestionados (en lugar de algún método específico de la implementación). En la cláusula 9 se da información detallada sobre los cambios que se han de realizar en los servicios de soporte del marco para el soporte de interfaces de granularidad gruesa.

El cambio más grande que se ha de efectuar en los servicios de soporte es en lo tocante al soporte para denominación. Las interfaces de fachada están vinculadas a nombres en el servicio de denominación, aproximadamente de la misma manera en que lo están las interfaces del servicio de soporte. No obstante, en las interfaces que utilizan fachadas no se requiere que los nombres de los objetos gestionados estén vinculados a las IOR en el servicio de denominación OMG. En cambio, se introduce un nuevo servicio como un lugar para almacenar nombres de objetos gestionados e información de relación de contenencia. Este nuevo servicio, el servicio de contenencia, se define

en 9.6. El servicio de contenencia estará soportado en sistemas que utilizan fachadas de objeto gestionado, pero no se requerirá en sistemas que no utilizan fachadas.

7.2.4 Modelado de fachadas

Para el soporte del esquema de diseño de fachada y la definición de fachadas utilizables con este marco, se introduce una nueva interfaz de base. Esta interfaz se denominará interfaz de fachada de objeto gestionado. Desempeña en las interfaces de granularidad gruesa el mismo papel desempeñado por la interfaz de objeto gestionado en las interfaces de granularidad fina. Así, es la interfaz de base de la cual deben heredar directa o indirectamente todas las interfaces de fachada de objeto gestionado para trabajar con el marco. La interfaz de fachada de objeto gestionado es muy similar a la interfaz de objeto gestionado definido en UIT-T X.780. Para la definición de la interfaz fachada de objeto gestionado véase UIT-T X.780.1, así como otras directrices sobre el modelado de información de granularidad gruesa.

En la figura 3 *infra* se muestran los cambios introducidos en el marco. Se ha añadido a la figura una nueva superclase, *ManagedObjectFacade*. Además se ha añadido el servicio de contenencia. Obsérvese que proporciona acceso 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 muestra con líneas de trazo discontinuo, lo que indica que no necesita almacenar los nombres y las IOR de objetos gestionados. No obstante, sigue siendo necesario el servicio de denominación para permitir que los sistemas de gestión encuentren interfaces fachada y referencias de servicio de soporte. Por último, los objetos gestionados también se muestran con líneas de trazo discontinuo, para indicar que no necesitan ser accesibles directamente.

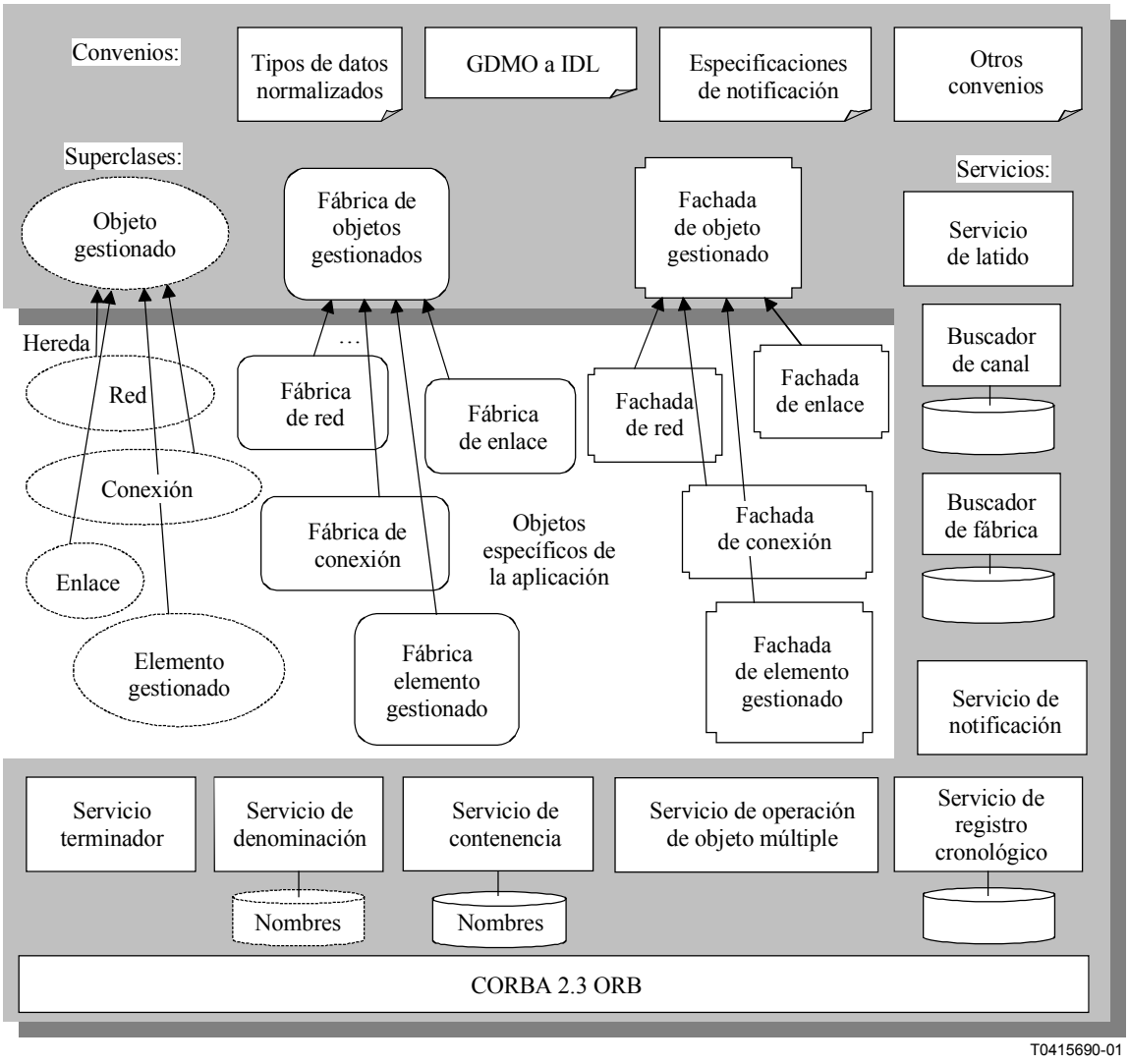

Figura 3/Q.816.1 – Marco con las extensiones para el soporte de interfaces de granularidad gruesa

8 Requisitos de utilización de los servicios de objetos comunes del marco para el soporte de interfaces de granularidad gruesa

En UIT-T Q.816 se describe la manera en que el marco original incluye varios de los servicios de objetos comunes OMG. Estos servicios están definidos por el OMG para uso, en general, en cualquier aplicación CORBA. El marco define cuáles de los servicios de objetos comunes del OMG debe soportar un sistema gestionado, y convenios para su utilización. En esta cláusula se proporcionan convenios y requisitos adicionales necesarios para el soporte de interfaces de granularidad gruesa para cada uno de los servicios de objetos comunes en el marco.

8.1 Servicio de denominación

En esta cláusula se describe la forma de utilizar el servicio de denominación OMG en interfaces de granularidad gruesa y la extensión de nombres de objetos gestionados necesaria para el soporte de fachadas.

8.1.1 Utilización del servicio de denominación en interfaces de granularidad gruesa

El cometido del servicio de denominación puede reducirse considerablemente en interfaces de granularidad gruesa. En interfaces de granularidad fina, los nombres de los objetos gestionados y las IOR se almacenan en el servicio de denominación. En interfaces de granularidad gruesa, los objetos gestionados no están obligados a exponer las IOR, y no es necesario almacenar sus vinculaciones de nombre en el servicio de denominación.

Sigue siendo aplicable el concepto de contexto de denominación raíz local. Recuérdese que un contexto de denominación de raíz local contiene vinculaciones de nombre para uno o más objetos, cada uno de los cuales constituye la raíz de un árbol de objetos gestionados sujetos a una relación de contenencia. Un contexto de denominación de raíz local, por sí mismo, tiene un nombre único, y todos los objetos por debajo del mismo tienen un nombre relativo a dicho contexto. Un contexto de denominación de raíz local contiene además vinculaciones para los servicios del marco que soportan los objetos gestionados que tienen nombres relativos a esa raíz local.

Las vinculaciones de nombre para los objetos gestionados no tienen que estar en el servicio de denominación, pero las vinculaciones para las interfaces de fachada sí deben figurar. Las interfaces de fachada sí tienen IOR, y son relativamente pocas, de manera que las vinculaciones de nombre para ellas sencillamente se colocan en el contexto de denominación de raíz local junto con las vinculaciones para los servicios de soporte. El formato de sus nombres se define más adelante.

El resultado de estas normas es que, en un sistema gestionado con una interfaz de granularidad gruesa, el servicio de denominación puede contener solamente contextos de denominación de raíz locales. Dado que muchos sistemas tendrán un solo contexto de denominación de raíz local, esos sistemas tendrán un solo objeto de contexto de denominación del servicio de denominación. Este objeto de contexto sólo contendrá vinculaciones de nombre para los servicios de soporte y las interfaces fachada de objeto gestionado implementadas por el sistema.

8.1.2 Nombres de interfaces de fachada

Según se ha mencionado anteriormente, las interfaces de fachada tienen IOR, y los nombres estarán vinculados a estas IOR en el contexto de denominación raíz. En esta cláusula se describe el formato de los nombres vinculados a las IOR de interfaz de fachada.

Puesto que los nombres para todas las fachadas en un sistema se colocan en el mismo contexto de denominación, cada uno debe ser único con respecto a cada uno de los demás. Además, es conveniente que un sistema de gestión pueda reconocer un nombre como perteneciente a una fachada, distinto de los nombres de servicio de soporte también vinculados en el contexto de denominación raíz. Finalmente, según se describe en 6.2, un sistema de gestión debe poder determinar la fachada que se ha de utilizar a partir del nombre de objeto gestionado. Para satisfacer este último requisito se requiere cierta coordinación entre los nombres de las fachadas y los nombres de objetos gestionados.

Cuando una IOR está vinculada a un nombre en un contexto de denominación OMG, el nombre consta de dos cadenas, una cadena *ID* y una cadena *kind*. Todas las interfaces de fachada estarán vinculadas con un valor de la cadena *ID* igual a "Facade" y la cadena *kind* fijada a un valor único entre todos los nombres de fachada en ese sistema. Este valor se denominará un "ID de fachada". Sin embargo, el ID de fachada se coloca en el campo *kind* en lugar de en el campo *ID*, porque, según se verá más adelante, debe concordar con el valor en la cadena *kind* en un nombre de objeto gestionado. Se consideró que era menos confuso hacer concordar un valor *kind* con otro valor *kind* que colocar un ID en un campo *kind*.

Dado que todos los nombres de fachada tienen el mismo valor *ID,* "Facade" las cadenas *kind* tienen que ser únicas para que la vinculación de nombre sea única, como lo requiere el servicio de denominación OMG. El valor de la cadena *kind* lo escogerá el implementador del sistema gestionado. El valor real no es importante, pero según se describe más adelante, cada objeto

gestionado al que se accede a través de esa fachada tendrá el mismo valor de la cadena *kind* en el componente final de su nombre.

8.1.3 Extensión de nombre de objeto gestionado

Los objetos gestionados accesibles a través de una fachada tendrán nombres idénticos a los de los objetos gestionados que no lo son, excepto los valores de cadenas *ID* y *kind* en el componente final del nombre. La Recomendación UIT-T Q.816 expresa que los nombres de objetos gestionados deberán tener un componente de nombre "extra" añadido al final de sus nombres distinguidos absolutos. Se requiere que este componente de nombre final represente relaciones de contenencia en el servicio de denominación OMG. En UIT-T Q.816 se exige que la cadena *ID* en este componente final del nombre tenga un valor de "Object" y que la cadena *kind* sea nula. La ampliación del marco para que soporte interfaces de granularidad gruesa modifica estos requisitos para los objetos accesibles a través de una fachada.

Los objetos gestionados que son accesibles a través de una fachada tendrán un componente final del nombre con la cadena *kind* fijada al valor de la cadena *kind* en la vinculación de nombre de la fachada que se utiliza para acceder a ese objeto. Es decir, a la cadena *kind* se le da el valor del ID de fachada elegido para esa fachada. Véase 8.1.2 *supra* para más información sobre los nombres de fachada. Los objetos gestionados que son accesibles únicamente a través de una fachada tendrán un valor de cadena *ID* nulo en el componente final de sus nombres. Los objetos gestionados que son accesibles tanto directamente como a través una fachada tendrán un valor *ID* de "Object" y un valor *kind* igual al ID de fachada en su componente de nombre final.

Para ilustrar cómo los nombres de objetos gestionados pueden resolverse hasta obtener una IOR de fachada considérese un sistema de gestión de elementos (EMS) perteneciente a la empresa de telecomunicaciones XYZ Communications. Este EMS podría tener un contexto de denominación raíz con el nombre globalmente único "ems17.ems.xyz.com". Supóngase que la fachada del grupo de circuitos en ese sistema está vinculado al nombre "Facade.cp" en el contexto de denominación raíz, donde "Facade" es el valor de la cadena *ID* y "cp" es el ID de fachada colocado en la cadena *kind*. En tal situación, una instancia de un objeto gestionado de grupo de circuitos podría tener el siguiente nombre:

```
ems17\.ems\.xyz\.com/me1.ManagedElement/bay1.Equipment/shelf1.Equipment/
slot1.Equipment/cp1.Equipment/.cp
```

(Cuando los nombres CORBA, que constan de una secuencia de estructuras de datos, cada una de las cuales contiene una cadena denominada "ID" y una denominada "Kind", se representan en forma de cadena, la Especificación del servicio de denominación OMG [5] especifica que se representen con el siguiente formato:

$ID_1.Kind_1/ID_2.Kind_2/ID_3.Kind_3\ldots$

donde ID_1 representa el valor de la cadena *ID* en el primer componente del nombre, etc. Así, una barra ('/') separa los valores de un componente de los valores del siguiente, y un punto ('.') separa el valor de la cadena *ID*, en cada componente, de la cadena *kind*. La cadena *ID* precede a la cadena *kind*, y si esta última está vacía se elimina el punto. Si el valor de una cadena *ID* o *kind* contiene un punto, una barra o una barra inversa ('\'), dicho carácter se determina como tal insertando antes del mismo una barra inversa. Ésta es la razón por la cual aparecen barras inversas que preceden a los puntos en el nombre anterior.)

Dado este nombre, un sistema de gestión puede determinar que el objeto es accesible mediante una fachada porque el valor de la cadena *kind* en el componente final no está vacío. A continuación, el sistema de gestión puede hallar, en dos pasos, la IOR de la fachada para ese objeto. En primer lugar, el sistema de gestión debe comparar el nombre con los contextos raíz que tiene registrados para encontrar el que mejor concuerda. En este ejemplo, se trata del contexto raíz denominado "ems17\.ems\.xyz\.com". Una vez que ha determinado esto, sabe que el nombre de la fachada para el

objeto es "ems17\.ems\.xyz\.com/Facade.cp". Lo sabe porque todas las fachadas en ese sistema están vinculadas a ese contexto raíz, todas tienen una cadena *ID* con el valor "Facade" y la cadena *kind* debe concordar con el valor de la cadena *kind* en el componente final del nombre del objeto gestionado, "cp". Suponiendo que el sistema de gestión retiene las IOR para las fachadas en una memoria cache, puede proseguir e invocar una operación sobre el objeto utilizando la IOR de la fachada y el nombre del objeto. Obsérvese que el sistema de gestión no tiene que recuperar ni retener en memoria cache muchas IOR, ni consultar al sistema gestionado para obtener la IOR antes de cada operación.

Puesto que el nombre de un objeto depende hasta cierto punto de la manera en que se accede a él, cambiar la manera de acceder a un objeto tiene el efecto secundario de tener que cambiar su nombre. Así, por ejemplo, si un objeto que anteriormente era accesible solamente a través de una fachada se hace accesible directamente, probablemente mediante una elevación del nivel del soporte lógico, esta mejora debe incluir también la actualización del nombre del objeto. Asimismo, si un objeto que era accesible a través de una fachada se traslada a otra, se debe cambiar su nombre. Para cambiar el nombre de un objeto puede ser necesario suprimirlo y volver a crearlo. Debido a este grado de dificultad, se prevé que el cambio de la forma de acceder a un objeto ocurra con poca frecuencia.

8.1.4 Comparaciones de nombres de objetos

En el cuadro 1 se compara la forma en que se construyen los valores de nombres para diferentes modalidades (*kind*) de interfaces.

Cuadro 1/Q.816.1 – Comparaciones de nombres de objetos

Tipo de interfaz	¿Nombre vinculado en contexto de denominación raíz?	Penúltimo componente del nombre		Último componente del nombre	
		ID	Modalidad (Kind)	ID	Modalidad (Kind)
Servicio de soporte	Siempre	No aplicable – sólo un componente usado	No aplicable – sólo un componente usado	Valor único escogido por el sistema gestionado, como "ChannelFinder1"	El nombre de interfaz, con todo su alcance, del servicio, como "itut_q816:: ChannelFinder"
Fachada	Siempre	Igual que el anterior	Igual que el anterior	"Facade"	Un valor único entre las vinculaciones de nombre de fachada en el contexto raíz
Objeto gestionado no accesible a través de una fachada	Los objetos más elevados tendrán un contexto de denominación vinculado al contexto raíz	Valor dependiente de la aplicación elegida para denominar el objeto con relación a su progenitor – el "RDN"	El valor de la cadena constante *kind* en el módulo de denominación referido en el momento de la creación del objeto	"Object"	<empty>

Cuadro 1/Q.816.1 – Comparaciones de nombres de objetos

Tipo de interfaz	¿Nombre vinculado en contexto de denominación raíz?	Penúltimo componente del nombre		Último componente del nombre	
		ID	Modalidad (Kind)	ID	Modalidad (Kind)
Objeto gestionado accesible a través de una fachada solamente	No es obligatorio que tenga un nombre en el servicio de denominación, pero los objetos más elevados se denominarán directamente subordinados al contexto de denominación raíz	Igual que el anterior	Igual que el anterior	<empty>	El valor elegido por el sistema gestionado para la cadena *kind* utilizada en la vinculación de nombre para la fachada a través de la cual se accede a este objeto. Véase la célula que explica los valores de la cadena *kind* para las fachadas
Objeto gestionado accesible a través de una fachada y también directamente	Igual que el anterior	Igual que el anterior	Igual que el anterior	"Object"	Igual que el anterior

8.1.5 Almacenamiento de nombres de objetos gestionados en el servicio de denominación

Aunque no es obligatorio, algunos sistemas que utilizan fachadas pueden optar, de todas formas, por almacenar algunos o todos los nombres de objetos gestionado en el servicio de denominación OMG. Esto puede ser debido a que algunos o todos los objetos exponen interfaces CORBA directas, y en consecuencia tienen IOR que pueden vincularse a sus nombres.

Si un objeto gestionado tiene una IOR vinculada a su nombre en el servicio de denominación y es también accesible a través de una fachada, su vinculación de nombre en el servicio de denominación tendrá un *ID* de "Object" y la cadena *kind* estará fijada al ID de fachada para su fachada.

Un sistema que almacena sólo algunos de sus nombres de objetos gestionados en el servicio de denominación tal vez tenga que resolver el problema de vincular una IOR a un nombre de objeto gestionado en el árbol de denominación debajo de un objeto gestionado que no tiene una IOR. Obsérvese que esto no será lo normal, ya que los objetos en las "hojas" del árbol de denominación suelen ser las más numerosas, por lo que es muy probable que no tengan IOR individuales. Los objetos interiores en el árbol suelen ser los menos numerosos, por lo que es muy probable que tengan IOR cuando se elija una implementación mixta de granularidad fina y gruesa. No obstante, si surge esta situación, se tratará creando un contexto de denominación para el objeto superior sin una IOR vinculada al valor de ID "Object". Por debajo de éste, el sistema creará un contexto para el nuevo objeto gestionado y vinculará su IOR al valor de ID "Object" en ese contexto. A la cadena *kind* se le dará el valor de la cadena *kind* vinculada a la fachada utilizada para acceder al objeto.

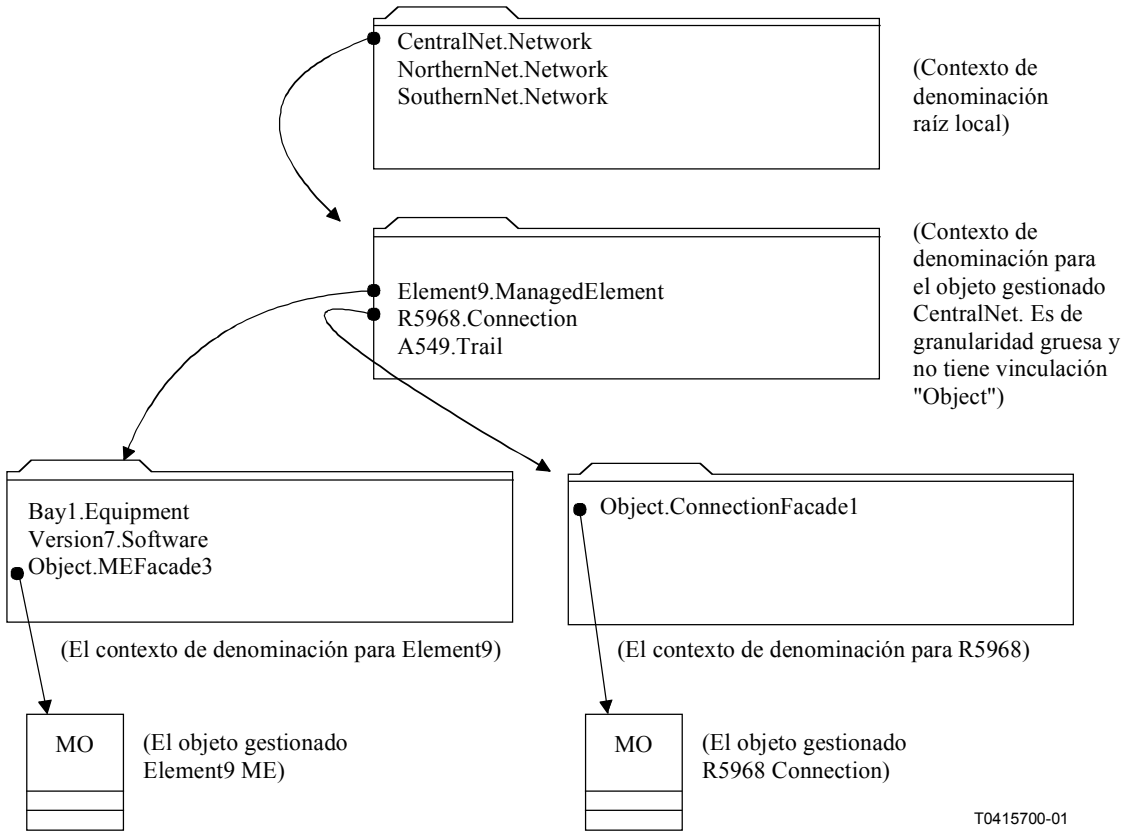

Figura 4/Q.816.1 – Gráfico de denominación que combina objetos de granularidad fina y gruesa

En este ejemplo, el objeto ManagedElement denominado "Element9" y el objeto Connection denominado "R5968" están ambos contenidos por el objeto Network denominado "CentralNet". CentralNet es accesible a través de una fachada, pero no directamente. Element9 y R5968 son accesibles tanto directamente como a través de fachadas. Para representar esto en el servicio de denominación se crea un contexto de denominación para CentralNet (en la figura aparece como la segunda carpeta desde arriba) pero no tiene vinculación para una IOR a CentralNet, ya que CentralNet no tiene una IOR directa. Si CentralNet sí tuviera una interfaz CORBA directa, el contexto de denominación CentralNet contendría una vinculación a dicha interfaz, con un valor de ID igual a "Object". Tanto Element9 como R5968 tienen vinculaciones para sus IOR, que se muestran en la parte inferior de la figura.

Puesto que no se requiere que los sistemas que soportan fachadas expongan IOR para objetos gestionados o vinculen nombres a IOR para objetos gestionados en el servicio de denominación, un sistema de gestión que utiliza una interfaz de granularidad gruesa no puede contar con el servicio de denominación para disponer de una información completa sobre la relación de contenencia. Un sistema gestionado puede que no vincule ningún nombre o que vincule solo algunos nombres a las IOR en el servicio de denominación. Por tanto, en las interfaces que soportan fachadas, los sistemas de gestión utilizarán el servicio de contenencia para recuperar información de contenencia. El servicio de denominación sólo puede utilizarse para resolver nombres hasta obtener las IOR para aquellos objetos que tienen IOR individuales.

Si se utiliza el servicio de denominación para almacenar nombres de objetos gestionados, incumbe al sistema gestionado mantener la información, en el servicio de denominación, exacta y sincronizada con la información en el servicio de contenencia. Un medio de conseguir esto podría ser implementar los dos servicios de modo que tuvieran acceso a una base de datos común. También podrían emplearse otros medios.

8.1.6 Requisitos del servicio de denominación para interfaces de granularidad gruesa

En esta cláusula se definen los requisitos relacionados con el servicio de denominación que deben cumplir los sistemas gestionados cuando utilicen interfaces de granularidad gruesa.

(R) NAME-1. Un sistema gestionado vinculará nombres a las IOR de cada fachada utilizada en el sistema. Una vinculación de nombre para cada fachada estará contenida en cada contexto de denominación raíz, donde el árbol bajo esa raíz contiene objetos accesibles a través de la fachada.

(R) NAME-2. La cadena *ID* de cada nombre de fachada tendrá el valor "Facade". La cadena *kind* tendrá un valor, elegido por el sistema gestionado, que deberá ser único entre las interfaces de fachada vinculadas en un determinado contexto de denominación raíz.

(R) NAME-3. Si se almacena en el servicio de denominación, el último componente (de nombre) del nombre de un objeto gestionado accesible a través de una fachada tendrá una cadena *ID* con el valor "Object" y una cadena *kind* con el mismo valor que la cadena *kind* utilizada en la vinculación de nombre para la fachada a través de la cual se accede al objeto gestionado.

(R) NAME-4. Si el sistema gestionado almacena algunos o todos los nombres de objetos gestionados en el servicio de denominación, el sistema gestionado asegurará la exactitud de esos nombres y la integridad del árbol de denominación.

(R) NAME-5. Si el sistema gestionado almacena algunos o todos los nombres de objetos gestionados en el servicio de denominación, utilizará contextos de denominación sin ninguna IOR vinculada al valor de *ID* "Object" para objetos gestionados que no tengan IOR individuales. Éstos sólo deben utilizarse cuando un objeto gestionado con una IOR está vinculado a un nombre en el árbol de denominación bajo un objeto gestionado sin IOR.

8.2 Servicio de notificación

No se establecen requisitos adicionales sobre la utilización del servicio de notificación para el soporte de interfaces de granularidad gruesa. El servicio de notificación puede utilizarse para transportar notificaciones desde objetos gestionados que son accesibles a través de una fachada de la misma manera que se utiliza para los objetos gestionados que no lo son.

8.3 Servicio de registro cronológico de telecomunicaciones

No se establecen requisitos adicionales sobre la utilización del servicio de registro cronológico de telecomunicaciones para el soporte de interfaces de granularidad gruesa.

8.4 Servicio de mensajería

No se establecen requisitos adicionales sobre la utilización del servicio de mensajería para el soporte de interfaces de granularidad gruesa.

8.5 Servicio de seguridad

Al utilizar el método de fachada, el servicio de seguridad OMG puede ofrecer protección solamente en el nivel de ejemplares de fachadas individuales. Si se requiere la protección de objetos gestionados en el nivel de ejemplar utilizando el servicio de seguridad OMG, debería utilizarse el método acceso de granularidad fina.

8.6 Servicio de transacción

No se establecen requisitos adicionales sobre la utilización del servicio de transacción para el soporte de interfaces de granularidad gruesa.

9 Requisitos de los servicios de soporte del marco para el soporte de interfaces de granularidad gruesa

Además de las normas para la utilización de servicios de objetos comunes OMG, la Recomendación UIT-T Q.816 define algunos nuevos servicios de soporte para uso en interfaces CORBA de la RGT. Estos servicios proporcionan funciones comunes específicas para la gestión de redes que no son proporcionadas por los servicios de objetos comunes OMG de uso general.

En las siguientes cláusulas se definen los requisitos adicionales que deben cumplir estos servicios para el soporte de interfaces de granularidad gruesa. Asimismo, se define un nuevo servicio para mantener relaciones de contenencia de recursos gestionados en interfaces de granularidad gruesa. El IDL que describe la interfaz para este nuevo servicio se adjunta en el anexo A.

9.1 Servicio buscador de fábrica

El servicio buscador de fábrica es un servicio definido en UIT-T Q.816 para permitir a los sistemas de gestión la búsqueda de "fábricas" en sistemas gestionados. Un sistema de gestión crea un nuevo objeto en un sistema gestionado invocando una operación sobre una fábrica, la cual, por sí misma, es un objeto. (Éste es un esquema de diseño CORBA comúnmente utilizado.) Un sistema de gestión busca una fábrica consultando a este servicio, prestado por el sistema gestionado, pasándole el nombre de clase de una fábrica. El servicio responde con una referencia a una fábrica de ese tipo. El sistema de gestión puede entonces invocar la operación apropiada sobre la fábrica y crear el objeto. El servicio buscador de fábrica se encuentra consultando al servicio de denominación. El sistema gestionado debe colocar una referencia a dicho servicio en el contexto de denominación raíz.

Este mismo método se utiliza para los objetos gestionados a los que se debe acceder a través de una fachada. La única diferencia estriba en que si el objeto no soporta una interfaz CORBA directa, la fábrica devolverá una referencia nula. Sin embargo, devolverá un nombre apropiado para que el sistema de gestión pueda acceder al objeto recién creado a través de su fachada.

Dado que el método para crear objetos gestionados en interfaces de granularidad gruesa es similar al empleado en el caso de interfaces de granularidad fina, no es necesario modificar el servicio buscador de fábrica. Por lo tanto, no se establecen requisitos adicionales sobre la utilización del servicio buscador de fábrica para el soporte de interfaces de granularidad gruesa.

9.2 Servicio buscador de canal

El servicio buscador de canal es un servicio definido en UIT-T Q.816 que permite a los sistemas de gestión averiguar qué canales de eventos están presentes en un sistema gestionado y qué notificaciones trata cada uno de ellos. Un sistema gestionado pequeño puede tener un solo canal, pero los sistema más complejos pueden tener múltiples canales, quizás para diferentes conjuntos de recursos gestionados o diferentes tipos de notificaciones. El servicio buscador de canal no permite que un sistema de gestión cree nuevos canales de eventos en el sistema gestionado, o cambie los eventos tratados por cualquiera de los canales. No obstante, canales de eventos individuales sí permiten que los sistemas de gestión añadan filtros y destinos para las notificaciones tratadas por el canal. Por lo tanto, el sistema gestionado registra información acerca de la configuración de los canales de eventos en el servicio buscador de canal, información que los sistemas de gestión pueden posteriormente recuperar y utilizar a fin de decidir qué canales habrán de percibirse. El servicio buscador de canal se encuentra consultando al servicio de denominación. El sistema gestionado debe colocar una referencia a dicho servicio en el contexto de denominación raíz.

Cuando el buscador de canal informe acerca de los objetos para los cuales trata eventos, identifica los objetos por su nombre. Puesto que los objetos gestionados en interfaces de granularidad gruesa tendrán nombres individuales sobre la base de la relación de contenencia al igual que los objetos gestionados en interfaces de granularidad fina, el buscador de canal puede usarse de la misma manera en las interfaces de granularidad gruesa. Por consiguiente, no se establecen nuevos requisitos

sobre la utilización del servicio buscador de canal para el soporte de interfaces de granularidad gruesa.

9.3 Servicio terminador

La Recomendación UIT-T Q.816 define el servicio terminador como un servicio común que realiza las funciones necesarias para suprimir un objeto gestionado. Cada objeto gestionado soportado por el marco tiene un atributo de política de supresión, que se fija cuando se crea el objeto. El servicio terminador garantiza que se sigue dicha política de supresión cuando se suprime un objeto. Además, los objetos gestionados se nombran sobre la base de relaciones de contenencia. Es por consiguiente importante que los objetos que contienen otros objetos solo se supriman si también se suprimen los objetos contenidos. El servicio terminador hace esto y garantiza la validez del árbol de denominación basado en la relación de contenencia.

Los objetos gestionados en una interfaz de granularidad gruesa se tratan de la misma manera. Por lo tanto, sigue siendo necesario que el servicio terminador desempeñe su función en interfaces de granularidad gruesa. No obstante, existen algunas diferencias en las interfaces de granularidad gruesa que afectan a la manera en que el servicio terminador desempeña su función. En primer lugar, si el servicio terminador utiliza la interfaz CORBA de objetos gestionados para recuperar la política de supresión de estos objetos (por oposición al caso en que se utiliza un medio de tipo privado), tendrá que hacerlo según la manera establecida en UIT-T X.780.1 para la interacción con objetos gestionados a través de una fachada. En segundo lugar, puesto que tal vez los nombres de objetos gestionados no estén almacenados en el servicio de denominación OMG, el servicio terminador tendrá que recuperar información de contenencia del servicio de contenencia (véase 9.6), y mantener la calidad correcta de la jerarquía de denominación en ese servicio. Si los nombres de objetos gestionados se almacenan en el servicio de denominación y también en el servicio de contenencia, el servicio terminador tendrá que actualizar los árboles de denominación de ambos servicios, a menos que los servicios se sincronicen ellos mismos.

Estos cambio, sin embargo, no requieren modificaciones de la interfaz IDL del servicio terminador. La Recomendación UIT-T Q.816 define esta interfaz con dos operaciones de supresión, una que toma un nombre de un objeto gestionado para identificar el objeto, y otra que toma una referencia de objeto gestionado (IOR). La operación que toma una referencia de objeto gestionado no es aplicable a las interfaces de granularidad gruesa, ya que cada uno de los objetos gestionados pueden no tener IOR únicas. La operación se define de manera que la referencia deba ser del tipo *ManagedObject* o una subclase. Las interfaces de fachada no heredan de esta interfaz, por lo que será imposible que un sistema de gestión trate de utilizar, por error, esta interfaz para suprimir un objeto gestionado que no tenía una IOR.

Para suprimir objetos gestionados en interfaces de granularidad gruesa, un sistema de gestión pasará el nombre del objeto a la operación *deleteByName* del servicio terminador, exactamente de la misma manera que lo haría para un objeto gestionado en un interfaz de granularidad fina. El servicio terminador reconocerá que el nombre hace referencia a un objeto accesible a través de una fachada sobre la base de la información que contiene el propio nombre (véase 8.1.3). Seguidamente el servicio terminador determinará la interfaz de fachada que puede utilizarse para acceder a ese objeto gestionado y recuperar la política de supresión del objeto, y hará seguir el nombre del objeto. Si el servicio terminador acaba suprimiendo el objeto, lo hará invocando la operación *destroy* en la misma fachada, y haciendo seguir de nuevo el nombre del objeto. Este proceso es muy similar al correspondiente en el caso de objetos gestionados en interfaces de granularidad fina, excepto que con estos objetos el servicio terminador utiliza la interfaz del objeto directamente, en lugar de una interfaz de fachada.

(R) TERM-1. El servicio terminador reconocerá las diferencias de nombre que existen entre los objetos que son accesibles a través una fachada y los objetos que no lo son, y accederá a ellos en consecuencia.

(R) TERM-2. Cuando se suprime un objeto gestionado accesible a través de una fachada y al que se accede utilizando CORBA, el servicio terminador utilizará la interfaz fachada del objeto para recuperar su política de supresión y destruirlo (a menos que utilice un método de acceso no basado en CORBA específico de la implementación).

(R) TERM-3. Cuando se suprime un objeto accesible a través de una fachada, el servicio terminador accederá a la información de contenencia almacenada en el servicio de contenencia. Asimismo, el servicio terminador se asegurará de que los nombres en el servicio de contenencia son correctos a medida que se suprimen los objetos.

9.4 Servicio de operación de objeto múltiple

La Recomendación UIT-T Q.816 define el servicio de operación de objetos múltiples (MOO) como un servicio común que un sistema de gestión puede utilizar para invocar operaciones sobre grupos de objetos gestionados con una sola (o pocas) invocaciones de método entre el sistema de gestión y el sistema gestionado. Para utilizar el servicio, el sistema de gestión invoca una sola operación en el servicio MOO residente en el sistema gestionado. Las operaciones soportadas son *get* (obtener), para recuperar uno o más valores de un grupo de objetos gestionados, *set* (fijar), para modificar uno o más valores en un grupo de objetos gestionados y *delete* (suprimir), para suprimir un grupo de objetos gestionados. El grupo de objetos sobre el que se opera se identifica con un alcance o ámbito (*scope*) y un filtro. Un alcance (o ámbito) identifica un objeto de base más algún conjunto de objetos contenido en ese objeto, fundándose en sus nombres. Un filtro es un enunciado lógico que prueba los valores de los atributos de los objetos en el ámbito de la operación. Si el resultado de la evaluación del enunciado es verdadero para un determinado objeto, la operación se aplica a ese objeto. Si la operación se aplica a muchos objetos, los resultados pueden ser demasiado grandes para ser retornados en un solo lote. El servicio MOO utiliza el esquema de diseño de iterador para permitir al sistema de gestión recuperar los resultados en un lote de tamaño manejable.

La Recomendación UIT-T X.780.1 también define una operación de fachada común que permite a un sistema de gestión recuperar valores de atributo de múltiples objetos accesibles a través de esa fachada, suministrando para ello una lista de nombres de objetos en una operación "bulk get" (obtener en masa). Sin embargo, esta operación "bulk get" no confiere la flexibilidad ni la potencia de las operaciones del servicio MOO. Es por consiguiente deseable que el servicio MOO también sea aplicable a las interfaces de granularidad gruesa.

No obstante, existen algunas diferencias en las interfaces de granularidad gruesa que afectan a la manera en que el servicio MOO desempeña su función. En primer lugar, dado que es posible que los nombres de objetos gestionados no estén almacenados en el servicio de denominación OMG, el servicio MOO tendrá que recuperar la información de contenencia del servicio de contenencia (véase 9.6). En segundo lugar, si el servicio MOO utiliza la interfaz CORBA de objetos gestionados (por oposición a un medio de tipo privado) para recuperar valores de atributo para el filtrado e invocar la operación real, tendrá que utilizar la fachada del objeto gestionado. En tercer lugar, si bien las interfaces de objeto de granularidad gruesa no disponen de métodos para retornar un nombre de objeto, el servicio MOO siempre retornará el nombre de cada objeto como parte de los resultados de una operación *get* confinada a un ámbito o alcance, como lo hace para las interfaces de granularidad fina. También debe permitir que los sistemas de gestión indiquen el nombre de los objetos en la lista de atributos que quieren recuperar.

Sin embargo, estos cambios no requieren que se introduzcan modificaciones en la interfaz IDL del servicio MOO ni en sus iteradores. La Recomendación UIT-T Q.816 define la interfaz del servicio MOO con tres operaciones: *get*, *set* y *delete*. Cada una de ellas toma el nombre del objeto de base, un indicador de ámbito y un enunciado de filtro. Los atributos sobre los que se actuará se identifican por el nombre. Todo esto es igualmente aplicable a las interfaces de granularidad gruesa que a las interfaces de granularidad fina. Lo que cambia en el caso de las interfaces de granularidad gruesa es

el lugar en que está situada la información de contenencia y la forma de acceso a los atributos de objetos gestionados.

La información de contenencia está situada en el servicio de contenencia y no en el servicio de denominación, y para acceder a los atributos de un objeto gestionado el servicio MOO tendrá que determinar la interfaz de fachada del objeto gestionado e invocar allí la operación, suministrando el nombre del objeto gestionado. La utilización del servicio de contenencia para recuperar la información de contenencia puede ser realmente más simple en las interfaces de granularidad gruesa ya que el servicio de contenencia acepta el nombre y el ámbito de un objeto de base. El servicio MOO sencillamente puede pasar al servicio de contenencia el nombre y el ámbito del objeto de base que se le ha suministrado y seguidamente iterar a través de los objetos contenidos para ver si pasan el filtro. No tiene que navegar por el árbol de denominación.

El acceso a atributos de objeto gestionado a través de una fachada no es muy diferente del acceso directo a atributos. Dado un nombre, la fachada correcta se determina haciendo concordar la "kind" de objeto (suministrado en el campo *kind* del último componente del nombre) con una fachada registrada en el contexto de denominación raíz. Seguidamente se aplica la operación a la fachada, suministrando el nombre del objeto gestionado como el primer parámetro en la operación.

(R) MOO-1. El servicio MOO reconocerá las diferencias que existen en los nombres de los objetos que son accesibles a través de una fachada de los objetos que no lo son, y los tratará en consecuencia.

(R) MOO-2. Cuando se accede a un objeto gestionado accesible a través de una fachada utilizando CORBA, el servicio MOO utilizará la interfaz fachada del objeto gestionado (a menos que utilice un método de acceso no basado en CORBA específico de la implementación).

(R) MOO-3. El servicio MOO determinará los objetos que están dentro del ámbito de una operación accediendo a la información de contenencia almacenada en el servicio de contenencia.

9.5 Servicio de latido

Debido a que CORBA proporciona transparencia con respecto a la ubicación, las aplicaciones de gestión de red pierden la visibilidad de las conexiones con sistemas específicos. Esto hace que las aplicaciones sean más fáciles de desarrollar, pero podría impedir que un sistema de gestión se diera cuenta de que se han perdido comunicaciones con un sistema gestionado. Para solucionar este problema, la Recomendación UIT-T Q.816 define el servicio de latido. Este servicio permite que un sistema de gestión configure un sistema gestionado para que emita periódicamente una breve notificación. La notificación pasa por cada uno de los canales de eventos del sistema gestionado, por lo que, además de probar el enlace de comunicaciones, prueba también la operación de los canales de eventos de notificación. Un sistema de gestión puede ser alertado de la existencia de problemas si no recibe una notificación de latido de un sistema gestionado cuando dicha notificación no se recibe a su debido tiempo.

Las interfaces de granularidad gruesa utilizarán canales de eventos de igual forma que las interfaces de granularidad fina. El servicio de latido se puede seguir utilizando para verificar la operación de estos canales y de la red de comunicaciones de datos que conecta el sistema gestionado con sus sistemas de gestión. No es necesario establecer requisitos adicionales sobre la utilización del servicio de latido para el soporte de interfaces de granularidad gruesa.

9.6 Servicio de contenencia

Las relaciones de contenencia entre objetos gestionados en interfaces de granularidad fina se representan por los nombres almacenados en el servicio de denominación OMG. Se requiere una capacidad similar para objetos gestionados en interfaces de granularidad gruesa. Es decir, se necesita una función que pueda informar qué objetos están contenidos en un objeto superior, que verifique la existencia de un objeto superior antes de crear un subordinado, que garantice que no se crean dos

objetos con el mismo nombre, etc. Para el soporte de esta función, se ampliará el marco añadiendo un nuevo servicio, el servicio de contenencia.

(R) CONTAINMENT-1. Un sistema gestionado ejemplificará al menos un objeto de servicio de contenencia. Asimismo, cada contexto de denominación raíz local en un sistema tendrá al menos una vinculación de nombre para un objeto de servicio de contenencia. El valor de la cadena *ID* en esta vinculación identificará sencillamente el servidor, quizás con un valor similar a "Containment1". La cadena *kind* en la vinculación identificará la clase del objeto ("itut_q816_1::ContainmentService").

9.6.1 Exposición razonada del servicio de contenencia

Hay muchos lugares diferentes donde podrían almacenarse nombres de objetos gestionados: el servicio de denominación, los objetos de fachada, un servicio existente como el servicio terminador o el servicio MOO, o un nuevo servicio.

No tiene realmente sentido poner los nombres en el servicio de denominación porque la verdadera finalidad del servicio de denominación es vincular nombres a las IOR, y unos objetos gestionados en interfaces de granularidad gruesa pueden no tener sus propias IOR. Vincular el nombre a la IOR de la fachada sería ineficaz. Todos los objetos gestionados accesibles a través de una misma fachada compartirían la misma IOR, por lo que todos sus nombres estarían vinculados a la misma IOR en el servicio de denominación. El resultado con toda probabilidad sería que en el servicio de denominación habría almacenadas grandes cantidades de copias de sólo unas pocas IOR.

Poner los nombres en los objetos de fachada, o los objetos gestionados propiamente dichos es una solución lógica, pero una de las finalidades del marco es proporcionar lugares para realizar funciones comunes. La duplicación de la gestión de la información de contenencia en cada fachada u objeto gestionado va en contra de esta finalidad.

El servicio terminador y el servicio MOO dependen de la información de contenencia para desempeñar sus tareas, por lo que una posibilidad es extender estos servicios para que gestionen la información de contenencia. Sin embargo, el principal cometido de estos servicios no es la gestión de los datos de contenencia. Si la gestión de la información de contenencia se coloca en cualquier otro lugar, estos servicios podrán seguir centrándose en las tareas para las cuales fueron concebidos.

Aunque añadir un nuevo servicio supone un cambio significativo en el marco, parece, en efecto, que es la mejor solución. Proporciona un depositario único para la información de contenencia y brinda una oportunidad para introducir formas de acceso a la información de contenencia que no están soportadas por el servicio de denominación, como utilizar el ámbito para indagaciones sobre objetos contenidos.

9.6.2 Descripción del servicio de contenencia

La función principal que ha de soportar el servicio de contenencia es permitir que un sistema de gestión consulte a un sistema gestionado con el nombre de un objeto y reciba en retorno los nombres de los objetos contenidos en ese objeto. Además, se definirá un medio que permita añadir nombres al servicio y suprimir nombres del servicio. Este medio no lo utilizarán los sistemas de gestión, sino internamente los objetos gestionados, fábricas y otras partes de un sistema gestionado. Se proporcionan para promover el desarrollo de componentes reutilizables, posiblemente por terceros, y se definen en una interfaz distinta de la utilizada por sistemas de gestión. Por último, en respuesta a una consulta es posible que se retornen grandes cantidades de nombres; por esta razón se utiliza el esquema de diseño de iterador. El iterador se describe en 9.6.2.3.

9.6.2.1 Interfaz del servicio de contenencia

El servicio de contenencia proporciona tres operaciones para recuperar información de contenencia. A continuación se presenta el IDL que describe la interfaz del servicio de contenencia (sin comentarios).

```
interface Containment {

     boolean exists (in NameType name)
               raises (ApplicationError);

     NameSetType getContained (in NameType base,
               in ScopeType scope,
               in unsigned short howMany,
               out NameIterator iterator)
               raises (ApplicationError);

     NameSetType getContainedByKind (in NameType base,
               in ScopeType scope,
               in KindType kind,
               in unsigned short howMany,
               out NameIterator iterator)
               raises (ApplicationError);

};   // end of Containment interface
```

La operación *exists* toma un nombre y retorna verdadero si dicho nombre está registrado en el servicio de contenencia. Las otras dos operaciones retornan los nombres de objetos contenidos por el objeto nombrado en el parámetro *base*. El parámetro *scope* en estas dos operaciones se puede utilizar para especificar qué parte del árbol de objetos contenidos bajo el objeto de base se ha de recuperar. La tercera operación, *getContainedByKind*, toma un parámetro *kind* para ordenar al servicio de contenencia que retorne los nombres para los objetos de cierta kind.

Una observación importante acerca de los nombres intercambiados a través de esta interfaz es que no es necesario que los nombres suministrados por el sistema de gestión en estas consultas contengan el componente final del nombre, donde la cadena *ID* es nula u "Object" y la cadena *kind* está vacía o fijada al identificador de fachada. No obstante, los nombres retornados al sistema gestionado siempre contendrán este componente final del nombre. Por lo tanto, si un sistema de gestión tiene el nombre de un objeto pero éste no contiene el componente final, puede obtenerlo consultando al servicio de contenencia para que busque los objetos contenidos pero con el ámbito (*scope*) igual a "sólo objeto de base". Un sistema de gestión podría encontrarse a sí mismo con un nombre de objeto al que le falta el componente final, por ejemplo, truncando un nombre de objeto para encontrar su progenitor.

Como se verá más adelante, antes de que los nombres puedan añadirse al servicio de contenencia, hay que registrar los nombre de los contextos de denominación raíz locales. Los contextos de denominación raíz locales registrados en el servicio de contenencia se pueden recuperar introduciendo un nombre *base* de longitud cero en a la operación *getContained* junto con un *scope* que indique el primer nivel de objetos contenidos. La introducción del nombre de un contexto de denominación raíz da lugar sencillamente a la recuperación de los nombres de los objetos gestionados vinculados por debajo de él. Esto mismo es válido para cualquier objeto gestionado.

(R) CONTAINMENT-2. La interfaz soportada por el (los) objeto(s) del servicio de contenencia deberá ser la interfaz de contenencia descrita anteriormente y definida en el IDL CORBA del anexo A.

(R) CONTAINMENT-3. En respuesta a una invocación de la operación *exists*, el servicio de contenencia retornará verdadero si el nombre está actualmente registrado en el servicio, y falso en caso contrario. Si se produce algún error en el servidor que impide realizar esta determinación, se formulará una excepción de error de aplicación apropiada.

(R) CONTAINMENT-4. En respuesta a una invocación de la operación *getContained*, el servicio de contenencia retornará una lista de los nombres de los objetos contenidos en el objeto nombrado en el parámetro *base*. La lista de objetos contenidos se determinará según el parámetro *scope* (véase la Recomendación UIT-T Q.816 para una descripción de la información de ámbito). El máximo número de nombres que habrán de ser retornados en el conjunto de resultados será el valor del parámetro *howMany*. Si hay que retornar más nombres que los indicados por *howMany*, el servicio de contenencia retornará una referencia a un iterador de nombre y pondrá a disposición el nombre adicional a través de esta interfaz. Si se pueden retornar todos los nombres en el conjunto de resultados, la referencia de iterador de nombre será nula. Si se introduce un nombre de base de longitud cero, el primer nivel de nombres contenidos estará constituido por los nombres de los contextos de denominación raíz registrados. Si algún error impide que se retorne la lista, se formulará una excepción de error de aplicación apropiada. En particular, si el nombre de base no está registrado, se formulará la excepción de error de aplicación *InvalidParameter*.

(R) CONTAINMENT-5. El servicio de contenencia responderá a la invocación de la operación *getContainedByKind* según se describe en el requisito CONTAINMENT-4, salvo que sólo se retornan aquellos nombres que concuerdan con el parámetro *kind*. Recuérdese que el componente final de un nombre tiene un *ID* de "Object". El *ID* del penúltimo componente es el que contiene el nombre distinguido relativo para el objeto, y el campo *kind* contiene el valor de la kind del objeto. Por lo tanto, en respuesta a una invocación de la operación *getContainedByKind*, el servicio de contenencia retornará sólo aquellos nombres en los que el campo *kind* del penúltimo componente del nombre concuerda con el valor del parámetro *kind*.

9.6.2.2 Interfaz de componente del servicio de contenencia

Algunas otras operaciones se definen en una interfaz distinta para uso interno por el sistema gestionado. Estas operaciones se definen para promover el desarrollo de componentes del servicio de contenencia reutilizables, posiblemente por terceros. A continuación se presenta el IDL que describe la interfaz de componente de servicio de contenencia (sin comentarios):

```
interface ContainmentComponent : Containment {

     void registerLocalRoot (in NameType name,
                             in NamingContext localRoot)
                    raises (ApplicationError);

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

     void addName (in NameType name)
                    raises (ApplicationError, CreateError);

     void removeName (in NameType name)
                    raises (ApplicationError, DeleteError);
};
```

Las dos primeras operaciones se utilizan para registrar y desregistrar contextos de denominación raíz locales en el servicio de contenencia. Cuando se añade un nombre al servicio de contenencia, el nombre de su objeto superior ya debe estar registrado; de lo contrario, el intento de añadir el nombre fracasará. En cambio, en la parte superior del árbol de denominación estarán los objetos que no tienen objeto superior en el sistema gestionado. Estos objetos se denominarán de forma relativa al contexto de denominación raíz local. Si no conoce cuáles son los nombres de los contextos de denominación raíz locales, el servicio de contenencia no puede saber si un nuevo nombre, con un objeto superior no reconocido, es incorrecto o forma parte de los objetos que están en la parte superior del árbol de denominación.

Las dos últimas operaciones se utilizan para añadir y suprimir nombres en el servicio de contenencia. Los clientes más probables de estas operaciones serán, respectivamente, las fábricas de objetos gestionados y el servicio terminador. Cuando se añade un nombre, se comprueba que el nombre es

único y que se ha añadido el nombre del objeto superior (a menos que se haya denominado directamente de forma relativa a un contexto de denominación raíz). Sólo se suprime un nombre si no hay ningún nombre subordinado que todavía esté registrado.

(O) CONTAINMENT-6. La interfaz soportada por el (los) objeto(s) del servicio de contenencia puede ser la interfaz de componente de contenencia descrita anteriormente y definida con el IDL CORBA del anexo A.

(O) CONTAINMENT-7. Cuando un contexto de denominación raíz local se registra utilizando la operación *registerLocalRoot*, el servicio de contenencia lo añadirá a su lista de contextos de denominación raíz locales registrados. Si un error en el servidor lo impide, se formulará una excepción de error de aplicación apropiada.

(O) CONTAINMENT-8. Cuando un contexto de denominación raíz local se desregistra utilizando la operación *unRegisterLocalRoot*, el servicio de contenencia lo suprimirá de su lista de contextos de denominación raíz locales registrados, pero solamente si no hay ningún nombre de objeto subordinado registrado. Si un error en el servidor lo impide, se formulará una excepción de error de aplicación apropiada.

(O) CONTAINMENT-9. Cuando se añade un nombre de objeto gestionado utilizando la operación *addName*, el servicio de contenencia añadirá dicho nombre a su lista de nombres de objetos registrados, solamente si el nombre es único y está directamente subordinado a una raíz local registrada o a otro nombre registrado. Si el nombre no es único o si no existe ningún nombre superior registrado, el servicio de contenencia no añadirá el nombre, sino que formulará la excepción de error apropiada. Obsérvese que si dos nombres son iguales salvo ciertas diferencias en el componente final del nombre (donde la cadena *ID* es "Object" y la cadena *kind* identifica una fachada), estos nombres no son únicos. Si no se puede realizar la operación debido a un error en el servidor, se formulará una excepción de error de aplicación apropiada.

(O) CONTAINMENT-10. Cuando se suprime un nombre de objeto gestionado utilizando la operación *removeName*, el servicio de contenencia suprimirá el nombre de su lista de nombres de objetos registrados, pero sólo si no hay nombres subordinados registrados. Si existen uno o más nombres subordinados, el servicio de contenencia no suprimirá el nombre, sino que formulará la excepción de error de supresión apropiada. Si no se puede realizar la operación debido a un error en el servidor, se formulará la excepción de error de aplicación apropiada.

9.6.2.3 Iterador de nombres

Cuando el servicio de contenencia retorna una lista de nombres de objetos contenidos es posible que la cantidad de nombres sea demasiado grande para que puedan ser retornados de una vez. Para que el servicio de contenencia pueda retornar una cantidad prácticamente ilimitada de nombres se utiliza el esquema de diseño de iterador. Como se ha descrito anteriormente, si el servicio de contenencia no puede retornar una lista de nombres en respuesta a una operación, retorna el mayor número posible junto con una referencia no nula a una interfaz de iterador. A continuación se presenta la descripción IDL de la interfaz de iterador de nombres.

```
interface NameIterator {

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

     void destroy();

}; // end of interface NameIterator
```

(R) CONTAINMENT-11. El servicio de contenencia soportará la utilización de iteradores de nombres con interfaces que concuerden con la descripción anterior y con la definición en el IDL del anexo A.

(R) CONTAINMENT-12. Cada vez que un cliente invoca una operación *getNext* en un iterador de nombres, éste retornará el siguiente conjunto de resultados. El iterador llevará la cuenta del número de resultados que el cliente ya ha recuperado y retornará todos los resultados de una vez. El iterador no retornará por segunda vez los resultados retornados inicialmente en respuesta a una operación en la interfaz del servicio de contenencia. En respuesta a una operación *getNext*, el iterador retornará, como máximo, el número de nombres indicados por el valor del parámetro *howMany*. El iterador puede retornar menos que el tamaño de lote solicitado, manteniendo así un equilibrio entre la eficiencia que se alcanza cuando los resultados se retornan en un solo lote de gran tamaño y la posible necesidad de bloquear hasta que haya más resultados disponibles. Si hay más resultados por devolver (además de los que se están devolviendo en ese momento), el valor de retorno de la operación *getNext* será verdadero y, en caso contrario, falso. El iterador no deberá devolver un conjunto de resultados vacío a menos que el parámetro *howMany* haya sido fijado a cero o no haya más resultados por devolver, lo que forzaría al cliente a interrogar al iterador.

(R) CONTAINMENT-13. El sistema gestionado controlará el ciclo de vida del iterador. Sin embargo, se proporciona una operación de destrucción por si el gestor quiere detener la recuperación de resultados antes de alcanzar la última iteración. Al invocar la operación *destroy*, el iterador liberará todos los recursos que esté utilizando y se suprimirá a sí mismo. Al retornar el último resultado, el iterador se destruirá a sí mismo. El iterador también puede ser destruido por el sistema gestionado si permanece sin funcionar durante un periodo de tiempo irrazonablemente largo.

10 Cumplimiento y conformidad

En esta cláusula se definen los criterios que deben ser satisfechos por otros documentos de normalización que pretendan cumplir este marco y las funciones que tienen que ser implementadas por sistemas que pretendan ser conformes con esta Recomendación.

10.1 Conformidad de sistemas

10.1.1 Puntos de conformidad

En esta cláusula se resumen las funciones individuales descritas anteriormente en esta Recomendación. Estos puntos de conformidad se combinan entonces en perfiles que deben ser soportados por los sistemas que pretendan ser conformes con esta Recomendación.

1) Servicio de denominación de granularidad gruesa: una implementación que pretenda ser conforme con los requisitos del servicio de denominación de granularidad gruesa deberá:
   - satisfacer todos los requisitos del servicio de denominación de granularidad gruesa especificados en 8.1.
2) Servicio terminador de granularidad gruesa: una implementación que pretenda ser conforme con los requisitos del servicio terminador de granularidad gruesa deberá:
   - satisfacer todos los requisitos del servicio terminador especificados en 9.3.
3) Servicio MOO de granularidad gruesa: una implementación que pretenda ser conforme con los requisitos del servicio MOO de granularidad gruesa deberá:
   - satisfacer todos los requisitos del servicio MOO especificados en 9.4.
4) Servicio de contenencia: una implementación que pretenda ser conforme con los requisitos del servicio de contenencia deberá:
   - satisfacer todos los requisitos obligatorios del servicio de contenencia especificados en 9.6. Los requisitos obligatorios están precedidos por "**(R)**". Los que están precedidos por "**(O)**" pueden estar soportados pero no son obligatorios.

10.1.2 Perfil básico de conformidad

Un sistema que pretenda la conformidad con el perfil básico Q.816.1 deberá soportar:

1) El perfil básico Q.816, **excepto**:
   - no se requiere que los nombres de objetos gestionados estén vinculados a las IOR de objetos gestionados en el servicio de denominación OMG.
2) Los requisitos del servicio de denominación de granularidad gruesa. (Véase el punto de conformidad 1 *supra*.)
3) Los requisitos del servicio terminador de granularidad gruesa. (Véase el punto de conformidad 2 *supra*.)
4) Los requisitos del servicio MOO de granularidad gruesa. (Véase el punto de conformidad 3 *supra*.)
5) Los requisitos de servicio de contenencia. (Véase el punto de conformidad 4 *supra*.)

10.2 Directrices para las declaraciones de conformidad

Los usuarios de este marco tienen que proceder con cuidado cuando formulen declaraciones de conformidad. Puesto que los módulos IDL se están utilizando como espacios de nombres, es posible, pues así lo autorizan las reglas IDL del OMG, que los módulos estén repartidos en distintos ficheros. Por lo tanto, cuando un módulo es extendido, su nombre no cambia. A diferencia de esto, un nuevo fichero IDL se añade simplemente. Por lo tanto, el simple enunciado del nombre de un módulo en una declaración de conformidad no bastará para identificar el conjunto de interfaces IDL. La declaración de conformidad tiene que identificar un documento y un año de publicación para garantizar que está identificada la versión correcta de IDL.

ANEXO A

IDL de servicio de soporte de granularidad gruesa

```
/* This IDL code is intended to be stored in a file named "itut_q816_1.idl"
located in the search path used by IDL compilers on your system. */

#ifndef ITUT_Q816_1_IDL
#define ITUT_Q816_1_IDL

#include <CosNaming.idl>
#include <itut_q816.idl>
#include <itut_x780.idl>

#pragma prefix "itu.int"


module itut_q816 {

// IMPORTED TYPES

     // Types imported from CosNaming
     typedef CosNaming::NamingContext NamingContext;
     typedef CosNaming::Istring KindType;

// INTERFACES

     /** The Name Iterator interface is used to retrieve the results from
     a Containment interface getContained or getContainedByKind operation
     using the iterator design pattern. */
```

```
interface NameIterator {

     /** This method is used to retrieve the next "howMany" results
     in the result set.
     @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 NameSetType results)
          raises (itut_x780::ApplicationError);

     /** This method is used to destroy the iterator and release its
     resources. The iterator, though, is automatically destroyed
     after the last results are returned, and may be destroyed if
     unused for an unreasonably long period. */

     void destroy();

}; // end of interface NameIterator


/** The Containment Service interface is used to retrieve from a
managed system information about the containment relationships between
the managed objects on the system. Containment relationships are
represented through names. An object that is contained by another
object is named relative to it. All managed object names are
registered with the Containment Service. */

interface Containment {

     /** This method is used to check to see if a name is registered
     with the Containment Service.
     @param name          The name to check to see if it is registered
     @return              True if the name is registered. All components
                          of the submitted name match a registered name.
     */

     boolean exists (in NameType name)
          raises (itut_x780::ApplicationError);

     /** This method is used to retrieve the names of the objects
     contained under a target object. An object is contained by the
     target object if its name begins with all but the last
     component of the target object's name. The iterator design
     pattern is used to support returning potentially large numbers
     of names. If the name does not exist, an application error
     exception is raised indicating an invalidParameter (See X.780
     for application error exception codes.)
     @param name          The name of the object for which the contained
                          objects are sought. Need not contain the final
                          name component (where ID=Object, kind = facade)
     @param scope         The scope is used to identify what part of
                          the tree of contained objects to return.
     @param howMany       The maximum number of names to return in the
                          results. If there are more than howMany names,
                          an iterator must be used to return the rest.
     @param iterator      A reference to an iterator to return additional
                          results. If all results are returned in
                          response to the call, this shall be null.
     @return              The names of the contained objects. Must
                          always contain the final name component.
     */
```

```
      NameSetType getContained (in NameType name,
            in ScopeType scope,
            in unsigned short howMany,
            out NameIterator iterator)
            raises (itut_x780::ApplicationError);

      /** This method is used to retrieve the names of the objects
      contained under a target object that match a specific kind.
      An object's kind is the value of the kind field in the
      next-to-last component of its name. This field must match
      the submitted kind value.
      @see getContained
      @param name          The name of the object for which the contained
                           objects are sought. Need not contain the final
                           name component (where ID=Object, kind = facade)
      @param scope         The scope is used to identify what part of
                           the tree of contained objects to return.
      @param kind          The value that must be matched in the kind
                           field of the next-to-last component in the
                           names returned.
      @param howMany       The maximum number of names to return in the
                           results. If there are more than howMany names,
                           an iterator must be used to return the rest.
      @param iterator      A reference to an iterator to return additional
                           results. If all results are returned in
                           response to the call, this shall be null.
      @return              The names of the contained objects. Must
                           always contain the final name component.
      */

      NameSetType getContainedByKind (in NameType name,
            in ScopeType scope,
            in KindType kind,
            in unsigned short howMany,
            out NameIterator iterator)
            raises (itut_x780::ApplicationError);

};    // end of interface Containment


/** The Containment Component interface extends the Containment
interface to add functions used internally to a managed system.
*/

interface ContainmentComponent : Containment {

      /** This method is used to register a local root naming context
      with the terminator service. Re-registering a name results in
      the newly supplied reference subsequently being used by the
      service, if it needs to access the root naming context. */

      void registerLocalRoot     (in NameType name,
                                  in NamingContext localRoot)
                                  raises (itut_x780::ApplicationError);

      /** This method is used to remove a local root naming context
      registration. */

      void unregisterLocalRoot (in NameType name)
            raises (itut_x780::ApplicationError);

      /** This method is used to add a name to the Containment
      Service. The name must be relative to an existing name or
      local root name. This means all but the last two name
      components of the submitted name must match all but the last
      name component of a registered name. Also, the name must
      be unique. This means that all but the last name component
      of the submitted name cannot match all but the last name
      component of any other registered name. If the name has
      no relative superior object registered, the name is not
```

```
            registered and a create error exception is raised with the
            cause set to badName. If the name is not unique, the name
            is not registered and a create error exception is raised
            with the cause set to duplicateName. (See X.780 for
            create error exeption codes.)
            @param name     The name to be added.
            */

            void addName (in NameType name)
                 raises (itut_x780::ApplicationError,
                      itut_x780::CreateError);

            /** This method is used to remove a name from the Containment
            Service. There must be no contained names registered with
            the Containment service when a name is removed. This means
            there must be no other names beginning with all but the last
            name component of the name to be removed. If there are, the
            name is not removed and a DeletError exception is raised with
            the cause set to containsObjects (see X.780). If the name
            does not exist, an application error is raised with the
            cause set to invalidParameter (See X.780).
            @param name     The name to be removed.
            */

            void removeName (in NameType name)
                 raises     (itut_x780::ApplicationError,
                             itut_x780::DeleteError);


      };    // end of interface ContainmentComponent

}; // end of module itut_q816

#endif // end of #ifndef ITUT_Q816_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

Impreso en Suiza
Ginebra, 2002