

Unión Internacional de Telecomunicaciones

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**M.3020**

(07/2011)

SERIE M: GESTIÓN DE LAS TELECOMUNICACIONES,  
INCLUIDA LA RGT Y EL MANTENIMIENTO DE REDES

Red de gestión de las telecomunicaciones

---

**Metodología para la especificación de interfaces  
de gestión**

Recomendación UIT-T M.3020

UIT-T



RECOMENDACIONES UIT-T DE LA SERIE M

**GESTIÓN DE LAS TELECOMUNICACIONES, INCLUIDA LA RGT Y EL MANTENIMIENTO DE REDES**

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

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

## Recomendación UIT-T M.3020

### Metodología para la especificación de interfaces de gestión

#### Resumen

La Recomendación UIT-T M.3020 describe la metodología para la especificación de interfaces de gestión (MISM), así como el proceso para obtener especificaciones de interfaces basadas en los requisitos, el análisis y el diseño (RAD) del usuario. Se proporcionan directrices para describir RAD utilizando la notación de lenguaje de modelización unificado (UML) aunque no se excluyen otras técnicas de especificación de interfaces. En esta Recomendación UIT-T figura una descripción de alto nivel de las directrices para utilizar UML.

#### Historia

Edición	Recomendación	Aprobación	Comisión de Estudio	ID único*
1.0	ITU-T M.3020	1992-10-05		<a href="http://handle.itu.int/11.1002/1000/1516-en">11.1002/1000/1516-en</a>
2.0	ITU-T M.3020	1995-07-27	4	<a href="http://handle.itu.int/11.1002/1000/1517-en">11.1002/1000/1517-en</a>
3.0	ITU-T M.3020	2000-02-04	4	<a href="http://handle.itu.int/11.1002/1000/4871-en">11.1002/1000/4871-en</a>
4.0	ITU-T M.3020	2007-07-22	4	<a href="http://handle.itu.int/11.1002/1000/9097-en">11.1002/1000/9097-en</a>
5.0	ITU-T M.3020	2008-07-29	4	<a href="http://handle.itu.int/11.1002/1000/9550-en">11.1002/1000/9550-en</a>
6.0	ITU-T M.3020	2009-05-14	2	<a href="http://handle.itu.int/11.1002/1000/9736-en">11.1002/1000/9736-en</a>
7.0	ITU-T M.3020	2010-09-06	2	<a href="http://handle.itu.int/11.1002/1000/10863-en">11.1002/1000/10863-en</a>
8.0	ITU-T M.3020	2011-07-14	2	<a href="http://handle.itu.int/11.1002/1000/11368-en">11.1002/1000/11368-en</a>

---

\* Para acceder a la Recomendación, sírvase digitar el URL <http://handle.itu.int/> en el campo de dirección del navegador, seguido por el identificador único de la Recomendación. Por ejemplo, <http://handle.itu.int/11.1002/1000/11830-en>.

## PREFACIO

La Unión Internacional de Telecomunicaciones (UIT) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones y de las tecnologías de la información y la comunicación. El Sector de Normalización de las Telecomunicaciones de la UIT (UIT-T) 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.

La observancia de esta Recomendación es voluntaria. Ahora bien, la Recomendación puede contener ciertas disposiciones obligatorias (para asegurar, por ejemplo, la aplicabilidad o la interoperabilidad), por lo que la observancia se consigue con el cumplimiento exacto y puntual de todas las disposiciones obligatorias. La obligatoriedad de un elemento preceptivo o requisito se expresa mediante las frases "tener que, haber de, hay que + infinitivo" o el verbo principal en tiempo futuro simple de mandato, en modo afirmativo o negativo. El hecho de que se utilice esta formulación no entraña que la observancia se imponga a ninguna de las partes.

## 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 en la dirección <http://www.itu.int/ITU-T/ipr/>.

© UIT 2014

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

## ÍNDICE

	<b>Página</b>
1 Alcance .....	1
2 Referencias .....	1
3 Definiciones.....	2
3.1    Términos definidos en otros documentos.....	2
3.2    Términos definidos en esta Recomendación .....	2
4 Abreviaturas.....	4
5 Convenios .....	4
6 Requisitos para la metodología y el soporte de notación .....	5
7 Metodología.....	5
7.1    Consideraciones generales.....	5
7.2    Aplicación y estructura de la metodología .....	5
7.3    Metodología detallada .....	5
8 Especificaciones de interfaces de gestión.....	8
9 Correlación en el proceso MISM.....	8
10 Estructura de la documentación.....	9
Anexo A – Requisitos .....	10
A.1    Convenios .....	10
A.2    Plantilla de requisitos .....	13
A.3    Plantilla de requisitos simplificada.....	16
Anexo B – Análisis .....	17
B.1    Convenios .....	18
B.2    Plantilla de análisis .....	20
B.3    Propiedades y herencia de IOC .....	29
Anexo C – Repertorio UML MISM.....	31
C.1    Introducción.....	31
C.2    Elementos de modelo básicos.....	31
C.3    Estereotipos .....	34
C.4    Clases de asociación .....	41
C.5    Clase abstracta .....	42
C.6    Aplicación de <<InformationObjectClass>> y <SupportIOC>> .....	43
Anexo D – Diseño.....	44
Anexo E – Definición de tipos de información – tipo repertorio.....	45
E.1    Tipos básicos .....	45
E.2    Tipo enumerado.....	45
E.3    Tipos complejos.....	45
E.4    Tipos útiles .....	46
E.5    Palabras clave .....	46

	<b>Página</b>
Anexo F – Directrices sobre importación de propiedades, herencia y entidad de IOC .....	47
F.1    Propiedad de IOC .....	47
F.2    Herencia.....	48
F.3    Importación de entidad (interfaz, IOC y atributo).....	48
Apéndice I – Ejemplo de requisitos .....	49
Apéndice II – Ejemplo de análisis .....	52
Apéndice III – Comparación con la Recomendación UIT-T Z.601 .....	60
Apéndice IV – Temas que quedan en estudio.....	61
IV.1    SOA .....	61
IV.2    UML .....	61
IV.3    Visibilidad .....	61
IV.4    Definición de tipos .....	61
Apéndice V – Ejemplos de utilización de UML adicionales .....	62
V.1    Clase proxy.....	62
Apéndice VI – Directrices sobre numeración de requisitos.....	64
Bibliografía .....	65

## Recomendación UIT-T M.3020

### Metodología para la especificación de interfaces de gestión

#### 1 Alcance

En esta Recomendación se describe la metodología para la especificación de interfaces de gestión (MISM). Se describe el proceso para obtener especificaciones de interfaces máquina-máquina basadas en los requisitos, el análisis y el diseño (RAD) del usuario. Se proporcionan directrices para describir RAD utilizando la notación de lenguaje de modelación unificado (UML) aunque no se excluyen otras técnicas de especificación de interfaces. Esta Recomendación contiene las directrices para utilizar UML. Una especificación de interfaz trata de los servicios de gestión definidos en [UIT-T M.3200] y/o soporta los procesos de gestión definidos en la serie [UIT-T M.3050.x]. Una especificación de este tipo puede soportar uno o más servicios de gestión o parte de los mismos. Los servicios de gestión comprenden las funciones de gestión, que pueden equipararse a las definidas en [UIT-T M.3400] o a los procesos definidos en la serie [UIT-T M.3050.x], especializadas para adaptarse a una zona gestionada específica, o que pueden ser nuevas, según proceda.

Esta metodología puede aplicarse tanto a los gestores/agentes tradicionales de interfaces de gestión [UIT-T M.3010] como a los principios de la arquitectura orientada al servicio (SOA) adoptados para la arquitectura de gestión de las redes de la próxima generación [UIT-T M.3060].

#### 2 Referencias

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

- [UIT-T M.3010] Recomendación UIT-T M.3010 (2000), *Principios para una red de gestión de las telecomunicaciones.*
- [UIT-T M.3050.x] Recomendación UIT-T M.3050.x (2007), *Mapa de operaciones de telecomunicación mejorado.*
- [UIT-T M.3060] Recomendación UIT-T M.3060/Y.2401 (2006), *Principios para la gestión de redes de próxima generación.*
- [UIT-T M.3200] Recomendación UIT-T M.3200 (1997), *Servicios de gestión de red de gestión de las telecomunicaciones y sectores gestionados de las telecomunicaciones: Panorama general.*
- [UIT-T M.3400] Recomendación UIT-T M.3400 (2000), *Funciones de gestión de la red de gestión de las telecomunicaciones.*
- [UIT-T Q.812] Recomendación UIT-T Q.812 (2004), *Perfiles de protocolo de capa superior para las interfaces Q y X.*
- [UIT-T X.680] Recomendación UIT-T X.680 (2008) | ISO/IEC 8824-1:2008, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica.*

- [UIT-T X.681] Recomendación UIT-T X.681 (2008) | ISO/IEC 8824-2:2008, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de objetos de información.*
- [UIT-T X.722] Recomendación UIT-T X.722 (1992) | ISO/IEC 10165-4:1992, *Tecnología de la información – Interconexión de sistemas abiertos – Estructura de la información de gestión: Directrices para la definición de objetos gestionados.*
- [UIT-T Z.100] Recomendación UIT-T Z.100 (2007), *Lenguaje de especificación y descripción.*
- [OMG UML] OMG: *Unified Modelling Language Specification, Version 1.5.*

Puede encontrarse una lista de referencias no normativas en la Bibliografía.

### 3 Definiciones

#### 3.1 Términos definidos en otros documentos

En esta Recomendación se utilizan los siguientes términos de [UIT-T M.3010]:

- usuario;
- servicio de gestión;
- conjunto de funciones de gestión.

En esta Recomendación se utilizan los siguientes términos de [OMG UML]:

- diagrama de actividad;
- actor;
- asociación;
- clase;
- diagrama de clase;
- clasificador;
- diagrama de colaboración;
- composición;
- modelElement;
- diagrama secuencial;
- diagrama de estados;
- estereotipo;
- caso de uso.

En esta Recomendación se utiliza el siguiente término de [UIT-T M.3060]:

- punto de referencia.

#### 3.2 Términos definidos en esta Recomendación

En esta Recomendación se definen los siguientes términos:

**3.2.1 agente (*agent*):** Engloba un subconjunto bien definido de funcionalidades de gestión. Interactúa con los gestores a través de una interfaz de gestión. Desde el punto de vista del gestor, el comportamiento del agente sólo es visible a través de la interfaz de gestión.

NOTA – Se considera equivalente al IRPAgent [b-3GPP TS 32.150].



**3.2.2 clase de objeto de información (*information object class*):** Describe la información que puede transmitirse/utilizarse en las interfaces de gestión y se modeliza utilizando el estereotipo "Class" en el metamodelo UML. En el Anexo B puede encontrarse una definición formal de la clase de objeto de información y su estructura de especificación.

**3.2.3 servicio de información (*information service*):** Describe la información relacionada con las entidades (ya sean recursos de red u objetos de soporte) que se han de gestionar y la manera en que puede gestionarse la información en una determinada área funcional. Se definen servicios de información para todos los IRP.

NOTA – Se considera idéntica a la definición de servicio de información de [b-3GPP TS 32.150].

**3.2.4 tipo de información (*information type*):** Especificación del tipo de parámetros de entrada de las operaciones.

**3.2.5 punto de referencia de integración (*integration reference point*):** Concepto arquitectónico descrito como un conjunto de especificaciones para la definición de un determinado aspecto de la interfaz de gestión, que comprende una especificación de requisitos, una especificación de servicio de información y una o más especificaciones de conjuntos de soluciones.

NOTA – Se considera idéntica a la definición de IRP de [b-3GPP TS 32.150].

**3.2.6 objetivos de gestión (*management goals*):** Objetivos de alto nivel de un usuario que realiza actividades de gestión.

**3.2.7 interfaz de gestión (*management interface*):** Ejecución de capacidades de gestión entre un gestor y un agente, que permite a un único gestor utilizar múltiples agentes y a un único agente soportar múltiples gestores.

NOTA – Q, C2B/B2B e Itf-N (3GPP) son ejemplos de interfaces de gestión.

**3.2.8 rol de gestión (*management role*):** Define las actividades que se prevé realicen los sistemas o el personal operativo encargado de la gestión de las telecomunicaciones. Los roles de gestión se definen independientemente de otros componentes, es decir, los recursos de telecomunicaciones y las funciones de gestión.

**3.2.9 hipótesis de gestión (*management scenario*):** Una hipótesis de gestión es un ejemplo de interacciones de gestión de un servicio de gestión.

**3.2.10 gestor (*manager*):** Modeliza un usuario de agente(s) e interactúa directamente con el/los agente(s) utilizando interfaces de gestión.

Dado que el gestor representa a un usuario agente, da una visión clara de lo que se supone ha de hacer el agente. Desde el punto de vista del agente, el comportamiento del gestor sólo es visible a través de la interfaz de gestión.

NOTA – Se considera equivalente a IRPManager [b-3GPP TS 32.150].

**3.2.11 información de correspondencia (*matching information*):** Especificación del tipo de un parámetro (posiblemente en referencia a IOC o un atributo de IOC).

**3.2.12 especificación independiente del protocolo (*protocol-neutral specification*):** Define las interfaces de gestión que soportan las capacidades de gestión sin ocuparse del protocolo y la representación de información implicados o requeridos por, por ejemplo, CORBA y XML.

**3.2.13 especificación propia del protocolo (*protocol-specific specification*):** Define las interfaces de gestión que soportan las capacidades de gestión para una tecnología de gestión específica (por ejemplo, CORBA).

NOTA – Se considera equivalente al conjunto de soluciones [b-3GPP TS 32.150].

**3.2.14 recursos de telecomunicaciones (*telecommunications resources*):** Los recursos de telecomunicaciones son entidades físicas o lógicas que han de gestionarse utilizando servicios de gestión.

## 4 Abreviaturas

En esta Recomendación se emplean las siguientes abreviaturas:

3GPP	Proyecto asociado de tercera generación ( <i>3rd generation partnership project</i> )
ADM	Administrativo (uso: categoría de requisitos) ( <i>administrative</i> )
ASN.1	Notación de sintaxis abstracta uno ( <i>abstract syntax notation one</i> )
CM	Condicional-Obligatorio ( <i>conditional-mandatory</i> )
CO	Condicional-Optativo ( <i>conditional-optional</i> )
CON	Conceptual (uso: categoría de requisitos ) ( <i>conceptual</i> )
CORBA	Arquitectura de intermediario de petición de objeto común ( <i>common object request broker architecture</i> )
FUN	Funcional (uso: categoría de requisitos) ( <i>functional</i> )
GDMO	Directrices para la definición de objetos gestionados ( <i>guidelines for the definition of managed objects</i> )
IDL	Lenguaje de definición de interfaz ( <i>interface definition language</i> )
IOC	Clase de objeto de información ( <i>information object class</i> )
IRP	Punto de referencia de integración ( <i>integration reference point</i> )
IS	Servicio de información ( <i>information service</i> )
MISM	Metodología para la especificación de interfaces de gestión ( <i>management interface specification methodology</i> )
NA	No aplicable ( <i>not applicable</i> )
NE	Elemento de red ( <i>network element</i> )
NON	No funcional (uso: categoría de requisitos) ( <i>non-functional</i> )
OMG	Grupo de gestión de objetos ( <i>object management group</i> )
OO	Orientado a objetos ( <i>object oriented</i> )
OSI	Interconexión de sistemas abiertos ( <i>open systems interconnection</i> )
SDL	Lenguaje de especificación y descripción ( <i>specification and description language</i> )
SOA	Arquitectura orientada al servicio ( <i>service oriented architecture</i> )
SS	Conjunto de soluciones ( <i>solution set</i> )
TS	Especificación técnica ( <i>technical specification</i> )
UML	Lenguaje de modelización unificado ( <i>unified modelling language</i> )
XML	Lenguaje de marcación extensible ( <i>eXtensible markup language</i> )

## 5 Convenios

La cláusula A.1 contiene convenios aplicables a la fase de requisitos.

La cláusula B.1 contiene convenios aplicables a la fase de análisis.

## **6 Requisitos para la metodología y el soporte de notación**

Para elaborar la metodología y seleccionar una notación, se tuvieron en cuenta los siguientes objetivos:

- 1) La metodología, incluida la elección de la notación, debe soportar la captación de todos los requisitos pertinentes del espacio problema, es decir, la gestión de telecomunicaciones.
- 2) La metodología facilita la producción de requisitos, sus correspondientes servicios de análisis|información y sus correspondientes especificaciones de diseño|conjuntos de soluciones.
- 3) La notación facilitará la generación inequívoca de la especificación en el perfil de protocolo de gestión objetivo. La metodología no tiene en cuenta las posibles opciones de servicios de protocolo (por ejemplo, servicio de seguridad corba).  
NOTA – Los protocolos de gestión aplicables para su utilización en el UIT-T se especifican en [UIT-T Q.812].
- 4) La metodología permitirá la especificación de elementos obligatorios y optativos en las tres fases. También especifica la relación de los elementos obligatorios|optativos entre las tres fases.
- 5) Se deben poder generar, a partir de la especificación independiente del protocolo (Análisis|IS), definiciones específicas del lenguaje, es decir, Diseño|SS (por ejemplo, de UML a IDL, de UML a GDMO/ASN.1).

## **7 Metodología**

### **7.1 Consideraciones generales**

El objetivo de esta metodología es describir los procesos que conducen a la definición de interfaces de gestión máquina-máquina.

### **7.2 Aplicación y estructura de la metodología**

La metodología para la especificación de interfaces de gestión (MISM) especifica un proceso en tres fases con características que permiten la correlación a lo largo de las mismas. En las tres fases se utilizan técnicas aceptadas por la industria que utilizan los principios de diseño y análisis orientado a objetos. Estas tres fases son: requisitos, análisis y diseño. Las técnicas deben permitir la utilización o creación de herramientas de soporte comercializadas. Pueden emplearse diversas técnicas en cada fase, en función de la naturaleza del problema.

### **7.3 Metodología detallada**

#### **7.3.1 Aspectos generales**

Las fases de requisitos y de análisis producen especificaciones UML. La fase de diseño utiliza una notación específica del paradigma de gestión de red. Los resultados de las tres fases son:

- Fase de requisitos – Requisitos.
- Fase de análisis – Especificación independiente de la aplicación.
- Fase de diseño – Especificación específica de la tecnología.

En un primer momento, la fase de diseño se llevará a cabo utilizando un manual o un enfoque personalizado. Cuando las herramientas puedan generar una definición específica del protocolo compatible, podrá aplicarse la notación UML a la fase de diseño.

En las siguientes cláusulas se describen las tres fases.

### 7.3.2 Requisitos

Los requisitos relativos al problema que debe resolverse corresponden a dos clases. En este contexto, la primera clase de requisitos se denomina requisitos comerciales. Un experto en la materia deberá estar en condiciones de determinar que los requisitos representan adecuadamente las necesidades del problema de gestión que tiene que resolverse. La segunda clase de requisitos son los requisitos de especificación. Estos requisitos proporcionarán los detalles suficientes para que pueda elaborarse la definición de interfaz en las fases de análisis y diseño. Como las definiciones finales de la interfaz deben ser correlacionables con los requisitos, puede ser necesario un proceso interactivo entre las tres fases. Este proceso interactivo tendrá que resolver cualquier ambigüedad en los requisitos a fin de asegurar la elaboración de una especificación que pueda llevarse a cabo.

Los datos de la interfaz persona-máquina pueden especificarse en la segunda clase de requisitos. Estos requisitos podrán tener una mayor repercusión en los conceptos y datos diseñados en las fases subsiguientes. Pueden encontrarse más detalles al respecto en el Apéndice III. Véase la serie de Recomendaciones UIT-T M.1400 sobre el diseño de datos para interfaces persona-máquina.

Para definir ambas clases de requisitos pueden utilizarse distintas técnicas. Independientemente de la técnica, la legibilidad de los requisitos es fundamental. No es necesario que los requisitos tengan una notación legible por la máquina mientras sean posibles la legibilidad y la correlación. Se recomienda la enumeración de requisitos para delinear los distintos requisitos a efectos de correlación.

Entre los aspectos identificados en la fase de requisitos figuran la política de seguridad, el ámbito de aplicación del dominio problema con respecto a las aplicaciones, los recursos y cometidos asumidos por los recursos. Los requisitos definen cometidos, responsabilidades, así como relaciones entre las entidades constituyentes del espacio problema. Para especificar los requisitos comerciales, pueden utilizarse distintas técnicas, incluida la representación textual. A fin de facilitar la correlación de estos requisitos con las fases de diseño y aplicación, se recomienda la enumeración de requisitos.

El problema debe limitarse a un ámbito de aplicación específico. Los servicios de gestión definidos en [UIT-T M.3200] y el conjunto de funciones definidas en [UIT-T M.3400] constituyen una forma de determinar el ámbito de aplicación. Los requisitos se especifican con los recursos que se gestionan y las funciones de gestión. En [UIT-T M.3050.x], "Mapa de operaciones de telecomunicación mejorado", donde se ofrece un enfoque basado en el proceso comercial, puede encontrarse una alternativa al método de servicios de gestión.

La relación entre [UIT-T M.3200] y [UIT-T M.3050] se describe en [UIT-T M.3050.x].

Las funciones de gestión deben agruparse y soportarse en aplicaciones que encaren necesidades comerciales específicas, por lo que el enlace entre los procesos eTOM y los conjuntos de funciones de gestión [UIT-T M.3200] y las funciones de gestión [UIT-T M.3400] es importante para ayudar a dar claridad y hacer efectivo este agrupamiento. Para satisfacer los requisitos comerciales del problema puede ser necesaria una ampliación de [UIT-T M.3400].

Deben utilizarse casos de utilización y escenarios UML para interactuar con los expertos en la materia al captar los requisitos comerciales. Los requisitos identificarán también las condiciones de fallo visibles en el proceso comercial.

NOTA – No se requiere la expresión de todos los requisitos como casos de uso.

Los requisitos producidos deben estar completos y ser detallados. Para lograrlo, debe utilizarse la naturaleza recurrente de la metodología. El carácter integral de los requisitos (claros y adecuadamente documentados) orienta las fases análisis y diseño.

En la cláusula A.1.2 se encuentran las directrices y la plantilla de identificación y estructura de requisitos.

Los casos de uso son objetivos que se logran siguiendo una serie de etapas. Cada etapa puede considerarse un subobjetivo del caso de uso. Así, cada una de esas etapas representa otro caso de uso (caso de uso subordinado) o una acción autónoma en el nivel inferior de descomposición del caso.

En la cláusula A.1.2 se encuentran las directrices y la plantilla para casos de uso.

Puede encontrarse un ejemplo de definición de requisitos en el Apéndice I.

### **7.3.3 Análisis**

En la fase de análisis se utilizan requisitos para identificar las entidades que interactúan, sus propiedades y las relaciones que mantienen entre sí. De esta forma, se pueden definir las interfaces ofrecidas por las entidades. En la notación UML, estas entidades pasan a ser clases. Las descripciones de clases junto con las interfaces expuestas deben correlacionarse con los requisitos. La relación entre las clases, definidas en la especificación de análisis, y las clases en la especificación de diseño no es necesariamente una relación de uno a uno.

En esta fase se han de tener en cuenta las necesidades de datos de interfaz persona-máquina (es decir, el modelo de información debe contener información suficiente para poder elaborar un diseño basado en los resultados del análisis).

En esta Recomendación figuran orientaciones de alto nivel relativas a la utilización de la notación UML destinada a soportar la especificación de interfaces de gestión, aunque para aumentar las definiciones UML podría utilizarse SDL [UIT-T Z.100].

La fase de análisis debe ser independiente de las condiciones aplicadas al diseño. Por ejemplo, el análisis puede documentarse con principios orientados al objeto aunque el diseño puede utilizar una tecnología no orientada al objeto. La información presentada en la fase de análisis comprende las descripciones de clase, las definiciones de datos, las relaciones de clases, los diagramas de interacción (diagramas secuenciales y/o diagramas de colaboración), los diagramas de transición de estados y los diagramas de actividad. Las definiciones de clase abarcan la especificación de operaciones, las notaciones, los atributos y los comportamientos presentados como notas o descripciones textuales.

Durante la fase de análisis de deben reutilizar servicios de gestión comunes independientes del protocolo (de haberlos), u otros servicios existentes, en pro de la armonización de las interfaces de gestión.

En el Anexo A se encuentran las directrices y la plantilla para casos de uso.

La plantilla de análisis utiliza el tipo de información como característica para describir los atributos IOC y los parámetros de operación/notificación. En el Anexo E se definen los tipos de información válidos que pueden utilizarse y su semántica.

### **7.3.4 Diseño**

#### **7.3.4.1 Aspectos generales**

En la fase de diseño se crea una especificación de interfaz compatible que puede ejecutarse. Esto implicará la selección de un lenguaje de especificación. Las especificaciones de la fase de diseño dependen del paradigma de gestión específico (por ejemplo, ILD para interfaces CORBA).

En esta fase se distinguen tres tipos de especificaciones de datos: diseño de datos dependientes del paradigma de gestión (por ejemplo, XML) que se han de comunicar por múltiples interfaces (por ejemplo, fallo y calidad de funcionamiento), mensajes (por ejemplo, informe de alarmas) que se han de comunicar por cada una de las interfaces, y método de codificación de los datos (por ejemplo, XML comprimido) coherente con un paradigma concreto.

La selección de un paradigma de gestión específico se aborda en otras Recomendaciones UIT-T, pero en las cláusulas siguientes se hace un breve repaso.

En la fase de diseño, se recomienda hacer referencia a las descripciones UML de las fases de requisitos y análisis para aumentar la especificación de comportamiento. Por ejemplo, la definición de comportamiento de GDMO puede hacer referencia a gráficos de estado, diagramas secuenciales y definiciones de clase en la fase de análisis. Si es necesario, pueden incluirse diagramas UML adicionales que describen interacciones entre entidades, correspondientes a determinados paradigmas de protocolo.

A medida que se adopten paradigmas adicionales para la gestión, se utilizarán las notaciones o los lenguajes definidos por estos paradigmas.

#### **7.3.4.2 CORBA**

En el contexto de la gestión basada en CORBA, el modelo de información se define con IDL.

#### **7.3.4.3 GDMO**

En el contexto del paradigma basado en la gestión de sistemas OSI [UIT-T X.722], la especificación de diseño es la especificación del modelo de información con plantillas GDMO para clases de objetos gestionados, atributos, comportamientos, notificaciones, acciones, denominación de instancias de la clase, y especificaciones de error/excepción. La sintaxis de la información se indica con la notación ASN.1 [UIT-T X.680].

En GDMO, la jerarquía de clases de objeto indica las propiedades de las clases de objeto necesarias para la gestión. Para aprovechar al máximo la reutilización de especificaciones, será necesario recurrir en gran medida a las relaciones de herencia (superclases y subclases). Las clases de objeto se indican utilizando las plantillas de [UIT-T X.722]. Las plantillas que definen el modelo de información deben registrarse (de conformidad con las reglas de [UIT-T X.722]) con un valor para el identificador de objeto ASN.1. Para las clases de objeto ya señaladas en otras Recomendaciones del UIT-T y de las normas ISO, sólo es necesario hacer referencia a la Recomendación y clase de objeto en cuestión. La denominación no forma parte de la jerarquía de clases de objeto, ni es la finalidad de ésta.

#### **7.3.4.4 XML**

Queda en estudio.

### **8 Especificaciones de interfaces de gestión**

La especificación de una interfaz de gestión comprende las especificaciones de requisitos, análisis y diseño citadas en la cláusula 7. En los Anexos A, B y C figura una estructura para especificar estas especificaciones.

Estas notaciones técnicas y de soporte se aplican también cuando se diseña un sistema para las especificaciones de interfaces de gestión, aunque no se considera que el diseño del sistema forme parte de las Recomendaciones UIT-T sobre gestión. Dichas notaciones contribuyen a describir la forma en que se aplican las especificaciones de interfaces al gestionar los recursos en un sistema como, por ejemplo, un elemento de red (NE, *network element*).

### **9 Correlación en el proceso MISM**

Para lograr la correlación entre los requisitos, el análisis y el diseño es necesario que se asigne la identificación apropiada. La correlación se efectúa mediante referencias entre las entidades especificadas en cada fase y entre ellas. La correlación se efectúa entre el diseño|conjunto de soluciones y los servicios de análisis|información y entre los servicios de análisis|información y los requisitos. La correlación puede efectuarse también entre artefactos de la especificación de

requisitos y entre artefactos de los servicios de análisis/información, por ejemplo, entre casos de uso y requisitos textuales. Los requisitos se han de identificar como se indica en la cláusula 7.3.2. Para los diversos casos de uso, el resultado de la fase de análisis indica requisitos de información más detallados. La fase de diseño debe indicar los diversos diagramas y textos del resultado de la fase de análisis. El indicador puede estar en función de una referencia a las cláusulas adecuadas.

La correlación entre la fase de diseño y los requisitos en cuestión suele ser indirecta. Esto se debe a que el resultado de las fases se define con distintos niveles de detalle.

En el Anexo B se encuentran directrices para la correlación entre la fase de requisitos y la fase de análisis.

Se recomienda el siguiente mecanismo para la correlación con los requisitos, etc., especificados en otros documentos (que posiblemente no se adapten al esquema de identificación recomendado):

foro/órgano ":" ID de documento ":" id

donde "id" puede ser uno de los siguientes:

- 1) ID de requisito;
- 2) ID de caso de uso;
- 3) título/texto del requisito;
- 4) título del caso de uso;
- 5) subcláusula del documento que identifica inequívocamente un requisito o caso de uso.

Ejemplos:

3GPP::32.111-1::getAlarmList

UIT-T::M.3016::1.5.1.2

## **10 Estructura de la documentación**

Aunque haya tres fases, la documentación de la interfaz puede combinar sus resultados en uno o más documentos. Se recomienda que para cada paradigma de protocolo de gestión de la red se combinen requisitos y análisis y se elaboren documentos de diseño separados.

## **Anexo A**

### **Requisitos**

(Este anexo forma parte integrante de la Recomendación.)

- A.1 *Convenios*
  - A.1.1 *Utilización de la notación UML para los requisitos*
  - A.1.2 *Plantilla de caso de uso*
  - A.1.3 *Categorías de requisitos*
- A.2 *Plantilla de requisitos*
  - 1 *Conceptos y antecedentes*
  - 2 *Requisitos comerciales*
    - 2.1 *Requisitos*
    - 2.2 *Roles de los actores*
    - 2.3 *Recursos de telecomunicaciones*
    - 2.4 *Casos de uso de alto nivel*
  - 3 *Requisitos de nivel de especificación*
    - 3.1 *Requisitos*
    - 3.2 *Roles de los actores*
    - 3.3 *Recursos de telecomunicaciones*
    - 3.4 *Casos de uso*
- A.3 *Plantilla de requisitos simplificada*
  - 1 *Conceptos y antecedentes*
  - 2 *Requisitos*

A continuación se dan las directrices para la especificación de requisitos. En el Apéndice I puede encontrarse un ejemplo de utilización de esta plantilla.

En la cláusula A.2 puede encontrarse la plantilla de requisitos normal (o formato completo). Además, en la cláusula A.3 se define una plantilla de requisitos simplificada.

#### **A.1 Convenios**

##### **A.1.1 Utilización de la notación UML para los requisitos**

En el Cuadro A.1 figura una comparación entre los conceptos de gestión y la notación UML. Esta Recomendación especifica los conceptos y notaciones de alto nivel que deben utilizarse en las distintas fases. Para ampliar la notación UML se utilizan estereotipos. En esta Recomendación (véase el Anexo C) figuran los estereotipos aprobados que se utilizan en el entorno de la gestión.



**Cuadro A.1 – Conceptos relacionados con los requisitos**

Concepto de gestión	Notación UML	Observaciones
usuario	Actor	Un usuario se modeliza como actor.
rol de gestión	Actor	Un actor desempeña un rol. Normalmente se aconseja modelizar únicamente un rol para cada actor.
función de gestión	caso de uso	Una función de gestión es modelizada por uno o más casos de uso.
conjunto de funciones de gestión	caso de uso	Un conjunto de funciones de gestión es un caso de uso compuesto con cada función de gestión (potencialmente) modelizada como un caso de uso separado.
servicio de gestión	caso de uso	Un servicio de gestión se modeliza como un caso de uso de alto nivel.
escenario de gestión	diagrama secuencial	Se prefieren diagramas secuenciales y no diagramas de colaboración.
tipo de recurso de telecomunicación	Clase	Los diagramas de clase describen los detalles de propiedad del tipo de recurso de telecomunicaciones en el nivel de detalle adecuado a la fase de la metodología.
objetivos de gestión	–	Los objetivos de gestión se presentan como descripciones textuales cuando no se aplica la notación UML.

### A.1.2 Plantilla de caso de uso

Cuando se faciliten casos de uso, se habrán de utilizar los siguientes convenios y plantillas.

**Cuadro A.2 – Plantilla para casos de uso**

Etapa del caso de uso	Evolución/especificación	Uso relacionado con <<Uses>>
Objetivo <sup>(*)</sup>	Se trata del objetivo/resultado final que quiere lograr el caso de uso y debe ser un enunciado conciso de lo que ha de lograr el caso de uso en el mejor de los casos. Puede haber una indicación de prioridad en relación con otros casos de uso y de rendimiento obligatorio del caso de uso, por ejemplo: <ul style="list-style-type: none"> <li>• tiempo real;</li> <li>• tiempo cuasi real;</li> <li>• tiempo no real.</li> </ul>	
Actores y roles <sup>(*)</sup>	Nombres de los actores/roles participantes en el caso de uso, incluidas las características de los roles de cada actor.	
Recursos de telecomunicaciones	Nombres de los recursos de telecomunicaciones participantes en el caso de uso.	
Supuestos	Descripción del entorno que sirve de contexto del caso de uso. Los supuestos y las condiciones <i>a priori</i> son mutuamente excluyentes. Los supuestos atañen a las propiedades estáticas.	

**Cuadro A.2 – Plantilla para casos de uso**

Etapa del caso de uso	Evolución/especificación	Uso relacionado con <<Uses>>
Condiciones <i>a priori</i>	<p>Lista de todas las condiciones del sistema y el entorno que han de ser verdaderas antes de poder activar el caso de uso.</p> <p>Las condiciones <i>a priori</i> y los supuestos son mutuamente excluyentes.</p> <p>Las condiciones <i>a priori</i> atañen a las propiedades dinámicas y pueden dar como resultado una excepción, lo que no ocurre nunca con los supuestos.</p>	
Inicio	<p>Nombre del evento que desencadena el caso de uso.</p> <p>Optativo; normalmente no se utiliza para especificar eventos tales como "cuando el administrador deba contener la información".</p>	
Etapa 1 <sup>(*)</sup> (M O)	<p>Un caso de uso describe una lista de etapas (manuales y automatizadas) necesarias para lograr el objetivo del caso de uso.</p> <p>Las etapas pueden invocar otros casos de uso.</p> <p>Las etapas se numeran para facilitar la correlación.</p> <p>Cada etapa se identifica como obligatoria (M) u optativa (O).</p> <p>Las subetapas se identifican en relación con la etapa que les corresponde, por ejemplo:</p> <p>Etapa n Etapa n.1 Etapa n.2 donde n.1 y n.2 son subetapas de la etapa n.</p>	Referencia a un caso de uso ya utilizado.
Etapa n (M O)	Se añaden las etapas necesarias en secuencia lógica.	
Fin <sup>(*)</sup>	<p>Lista de evento(s) que señala(n) que se ha completado el caso de uso.</p> <p>NOTA – En este contexto, "evento" ha de interpretarse en su sentido más general y no limitarlo a, por ejemplo, notificaciones intercambiadas a través de una interfaz de gestión. Por ejemplo, la compleción de un procesamiento puede considerarse como un evento que indica que se ha completado el caso de uso.</p>	
Excepciones	Lista resumida de condiciones de excepción y fallos detectados por el caso de uso durante su ejecución.	
Condiciones <i>a posteriori</i>	Lista de todas las condiciones del sistema y el entorno que han de ser verdaderas cuando se ha completado el caso de uso. El enunciado de condiciones <i>a posteriori</i> determina si se espera que el caso de uso sea un éxito completo, un éxito parcial o, incluso, un fracaso para considerarse completado.	
Correlación <sup>(*)</sup>	Requisitos o casos de uso revelados por el caso de uso.	
<p>NOTA – Los campos marcados con "*" son obligatorios para todas las especificaciones de casos de uso. Los demás campos sólo son obligatorios cuando son pertinentes para un caso de uso específico.</p>		

### A.1.3 Categorías de requisitos

Conviene clasificar los requisitos en diversas categorías. Las siguientes categorías se consideran pertinentes para la MISM:

- Conceptual (CON) – Identifica un concepto, tipo de datos, relación, formato o estructura.
- Funcional (FUN) – Identifica una capacidad funcional, situación dinámica, secuencia, parámetros de temporización o una interacción.
- No funcional (NON) – Requisitos no funcionales, incluidas las condiciones anormales, las condiciones de error y los límites del rendimiento.
- Administrativa (ADM) – Administración del sistema y requisitos operativos no relacionados con la ejecución normal de los casos de uso.

Los requisitos se han de escribir de acuerdo con la siguiente plantilla:

REQ-Etiqueta-Categoría-Número {Categoría, número} Detalles {Cita fuente }

donde "Etiqueta" es la abreviatura de la Recomendación (o de parte de la misma). El conjunto de etiquetas es infinito y no está sujeto a normalización.

En el Apéndice VI pueden encontrarse las directrices sobre numeración de requisitos.

## A.2 Plantilla de requisitos

### 1 Conceptos y antecedentes

*Define los objetivos y metas principales y las interfaces de gestión aplicables (y puntos de referencia) para esta especificación. Utiliza la clasificación por categorías de [UIT-T M.3200] como fuente de identificación de uno o más servicios de gestión soportados por esta interfaz.*

*En esta subcláusula se dará una descripción clara de las ventajas para los usuarios de la RGT, es decir del motivo para realizar este servicio de gestión. Se añadirán los antecedentes y el contexto, según proceda, pero las partes explicativa y descriptiva deben estar separadas. Cuando sea necesario, la información de apoyo sobre los antecedentes deberá figurar en un apéndice.*

#### 1.a SubClauseTitle

*SubClauseTitle es el nombre de la subcláusula.*

*"a" representa un número, empezando por 1 y aumentándose en 1 a cada nueva subcláusula.*

*La utilización de subcláusulas es optativa.*

### 2 Requisitos comerciales

#### 2.1 Requisitos

##### 2.1.a SubSetTitle

*SubSetTitle es el nombre de un subconjunto de requisitos comerciales.*

*"a" representa un número, empezando por 1 y aumentándose en 1 a cada nuevo subconjunto.*

*La utilización de subconjuntos es optativa y todos los requisitos comerciales pueden enunciarse en la subcláusula 2.1 (requisitos).*

*Se hace una lista con los principales requisitos en modo texto y se identifican los casos de uso con actores/roles y recursos. Los casos de uso de alto nivel (subcláusula 2.4 siguiente) han de revelar los requisitos comerciales y se distinguen de los requisitos de especificación por no afinar a niveles inferiores. En la cláusula 2.4 se presentan muchos ejemplos de lo que conforma un caso de uso de alto nivel. La información relacionada con las políticas (por ejemplo, seguridad, persistencia) reúne las condiciones necesarias para ser incluida en este nivel. Se recomienda la numeración de los requisitos a efectos de correlación*

*Los requisitos se han de especificar como se describe en la cláusula A.1.3. Dentro de la especificación de requisitos, se sugiere escribir estos últimos en la secuencia indicada en la cláusula A.1.3 (ya sea en toda la especificación o en cada uno de los subconjuntos).*

La utilización de categorías de requisitos es optativa y, en caso de utilizarse, puede recurrirse a subconjuntos de categorías.

Por ejemplo, el requisito conceptual número 23 de la Recomendación etiquetada como "SM" se especificará de la siguiente manera:

Identificador	Definición
REQ-SM-CON-23	Una Orden de Servicio consta de un nombre, una dirección, un número de teléfono, una descripción de servicio y un número de FAX optativo para la comunicación {T1M1.5 Documento 246 11/96}

Pueden utilizarse uno o más cuadros con texto explicativo entre ellos, según proceda.

## 2.2 Roles de los actores

Aquí se incluye una descripción textual del actor (véase la cláusula 3).

## 2.3 Recursos de telecomunicaciones

Se presenta la descripción textual de los recursos pertinentes (véase la cláusula 3) necesarios para soportar los casos de uso.

## 2.4 Casos de uso de alto nivel

Se presenta un diagrama de casos de uso de alto nivel. Para que los expertos en la materia comprendan el caso de uso, deben añadirse descripciones textuales para cada uno de esos casos. La descripción debe tener dos finalidades: captar el conocimiento de los expertos en dominios y validar los modelos en las fases análisis y diseño con respecto a los requisitos. En el Apéndice I figura un ejemplo de caso de uso de alto nivel.

### 2.4.a UseCaseName

UseCaseName es el nombre del caso de uso.

"a" representa un número, empezando por 1 y aumentándose en 1 a cada nueva definición de caso de uso.

Esta subcláusula se repite para cada caso de uso de alto nivel definido para los requisitos de la especificación de interfaz.

Los casos de uso de alto nivel pueden identificar los diversos conjuntos de funciones definidos en [UIT-T M.3400] o los procesos de gestión definidos en [UIT-T M.3050.x]. Estos casos de uso pueden llegar a ser más exactos, tal como se describe más adelante en la subcláusula requisitos de especificación, utilizando estereotipos tales como "include" (incluir) y "extend" (ampliar).

Podrán utilizarse diagramas secuenciales, cuando proceda. No obstante, no se prevé utilizar estos diagramas en requisitos de alto nivel. Estos diagramas pueden ser más adecuados cuando los casos de uso en ese nivel se descomponen en el nivel de requisitos siguiente.

La correlación del nivel de requisitos siguiente desde este nivel puede identificarse por la forma en que cada conjunto de funciones se vuelve más exacta con los nuevos casos de uso.

Utilizando la plantilla del Cuadro A.2, puede utilizarse un conjunto de cuadros de casos de uso para representar las capacidades importantes estudiadas al nivel de abstracción conveniente al problema que se analiza.

El nivel de detalle y el alcance de la cobertura de los casos de uso dependen de la familiaridad del equipo autor con el tema tratado y, por lo tanto, son subjetivos. Los niveles de detalle inferiores son más una indicación de análisis que una captación de requisitos.

Se pueden elaborar sucesivamente análisis más detallados de cada etapa de un caso de uso a niveles de abstracción superiores haciendo referencia al caso de uso más detallado en la celda del cuadro reservada a tal efecto. Cabe subrayar que esto último no es obligatorio, pero sí subjetivo, pues depende de las necesidades del autor/grupo.

A continuación se presenta una lista cuyo objetivo es ayudar en la identificación de los casos de uso adecuados:

- *Cuál es el principal objetivo del sistema*
- *Qué tipos de personas/sistemas han de interactuar con el sistema*
- *Cómo pueden estas personas/sistemas agruparse o reducirse a roles*
- *Cuáles son los aspectos de inicio, funcionamiento normal, fallo y recuperación del sistema*

- *Qué tipos de informes o datos se pueden necesitar del sistema*
  - *Qué actividades necesarias se requieren (por ejemplo, en función de la hora y la carga de la red).*
- Conviene documentar los casos de uso de manera común. Se sugiere la siguiente estructura:*
- *<cuadro de caso de uso > (véase el Cuadro A.2)*
  - *<diagrama(s) secuencial(es) optativo(s) >*
  - *<gráfico(s) de estados optativo(s) >.*

### **3 Requisitos de nivel de especificación**

#### **3.1 Requisitos**

*Los requisitos comerciales son más exactos cuando utilizan las funciones de gestión que figuran en [UIT-T M.3400]. Dado que [UIT-T M.3400] no es suficientemente exhaustiva para tratar todos los servicios de gestión de todas las áreas gestionadas, se prevé que serán necesarias nuevas funciones. Estas nuevas funciones deben incluirse en los requisitos como se indica a continuación.*

##### **3.1.a SubSetTitle**

*SubSetTitle representa el nombre de un subconjunto de requisitos de nivel de especificación.*

*"a" representa un número, empezando por 1 y aumentándose en 1 a cada nuevo subconjunto.*

*La utilización de subconjuntos es optativa y todos los requisitos de nivel de especificación pueden enunciarse en la subcláusula 3.1 (requisitos).*

*Enumerar los principales detalles y requisitos concretos en el texto e identificar los casos de uso con sus actores/roles y recursos. Los casos de uso de la subcláusula 3.4 deben revelar los requisitos de nivel de especificación con los detalles de menor nivel y estar más orientados a la aplicación en comparación con los requisitos de casos de uso comerciales. Se han de numerar los requisitos para la correlación.*

*Los requisitos han de especificarse como se describe en la cláusula A.1.3. En cada especificación de requisitos, se sugiere escribir los requisitos siguiendo la secuencia de la cláusula A.1.3 (para toda la especificación o para cada subconjunto).*

*La utilización de categorías de requisitos es optativa y, cuando así sea, puede aplicarse un subconjunto de categorías.*

*Por ejemplo, el requisito funcional número 33 de una Recomendación etiquetada 'OM' se especificará como sigue:*

<b>Identificador</b>	<b>Definición</b>
REQ-OM-FUN-33	El iniciador puede cancelar una operación pendiente

*Pueden utilizarse, según convenga, uno o más cuadros con texto explicativo entre ellos.*

*Los requisitos de nivel de especificación deben seguir los convenios y plantillas definidos en la cláusula A.1.*

#### **3.2 Roles de los actores**

*Se incluye aquí una lista de todos los actores y una descripción textual de los actores que no se hayan definido ya en los requisitos comerciales.*

#### **3.3 Recursos de telecomunicaciones**

*Se incluye aquí una lista de todos los recursos pasivos y una descripción textual de los recursos que no se hayan definido ya en los requisitos comerciales.*

#### **3.4 Casos de uso**

*Los casos de uso de alto nivel se hacen más concretos aquí utilizando varios casos de uso de nivel de especificación, cada uno de los cuales se explicará más detalladamente en una subcláusula, como se describe a continuación.*

### 3.4.a UseCaseName

*UseCaseName es el nombre del caso de uso.*

*"a" representa un número, empezando por 1 y aumentándose en 1 a cada nueva definición de caso de uso.*

*Si procede, pueden utilizarse diagramas secuenciales y de estados.*

*NOTA – Quedan en estudio las directrices y criterios para la utilización de diagramas secuenciales y diagramas de estados.*

*Las especificaciones de casos de uso han de seguir los convenios y plantillas definidos en la cláusula A.1.*

## A.3 Plantilla de requisitos simplificada

La plantilla de requisitos simplificada es una plantilla alternativa que se utilizará cuando sólo se necesitan los requisitos textuales. Se define una plantilla distinta para evitar las ambigüedades que resultarían de añadir opciones a la plantilla de formato completo de la cláusula A.2.

### 1 Conceptos y antecedentes

*Define los objetivos y metas principales y las interfaces de gestión aplicables (y puntos de referencia) para esta especificación. Utiliza la clasificación por categorías de [UIT-T M.3200] como fuente de identificación de uno o más servicios de gestión soportados por esta interfaz.*

*En esta cláusula se dará una descripción clara de las ventajas para los usuarios, es decir del motivo para realizar este servicio de gestión. Se añadirán los antecedentes y el contexto, según proceda, pero las partes explicativa y descriptiva deben estar separadas. Cuando sea necesario, la información de apoyo sobre los antecedentes deberá figurar en un apéndice.*

#### 1.a SubClauseTitle

*SubClauseTitle es el nombre de la subcláusula.*

*"a" representa un número, empezando por 1 y aumentándose en 1 a cada nueva subcláusula.*

*La utilización de subcláusulas es optativa.*

### 2 Requisitos

#### 2.a SubSetTitle

*SubSetTitle es el nombre de un subconjunto de requisitos comerciales.*

*"a" representa un número, empezando por 1 y aumentándose en 1 a cada nuevo subconjunto.*

*La utilización de subconjuntos es optativa y todos los requisitos comerciales pueden enunciarse en la cláusula 2 (requisitos).*

*Se hace una lista con los principales requisitos en modo texto y se identifican los casos de uso con actores/roles y recursos. El caso de uso permite determinar los requisitos de alto nivel y se distingue de los requisitos de especificación porque no logra ser más exacto en niveles más bajos. La información relacionada con ciertas políticas (por ejemplo, seguridad, persistencia) reúne las condiciones necesarias para ser incluida en este nivel. Se recomienda la numeración de los requisitos a efectos de correlación.*

*Los requisitos han de especificarse como se indica en la subcláusula A.1.3. En cada especificación de requisitos, se sugiere escribir los requisitos siguiendo la secuencia de la cláusula A.1.3 (para toda la especificación o para cada subconjunto).*

*La utilización de categorías de requisitos es optativa y, cuando así sea, puede aplicarse un subconjunto de categorías.*

*Por ejemplo, el requisito conceptual número 23 de una Recomendación etiquetada 'SM' se especificará como sigue:*

Identificador	Definición
REQ-SM-CON-23	Una Orden de Servicio consta de un nombre, una dirección, un número de teléfono, una descripción de servicio y un número de FAX optativo para la comunicación {TIM1.5 Documento 246 11/96}

*Pueden utilizarse uno o más cuadros con texto explicativo entre ellos, según proceda.*

## **Anexo B**

### **Análisis**

(Este anexo forma parte integrante de la presente Recomendación.)

- B.1 Convenios*
  - B.1.1 Calificativos obligatorios, optativos y condicionales*
- B.2 Plantilla de análisis*
  - 1 Conceptos y antecedentes*
  - 2 Clases de objetos de información*
    - 2.1 Entidades de información importadas y etiquetas locales*
    - 2.2 Diagrama de clase*
      - 2.2.1 Atributos y relaciones*
      - 2.2.2 Herencia*
    - 2.3 Definiciones de clases de objetos de información*
      - 2.3.a InformationObjectClassName*
    - 2.4 Definiciones de relaciones de información*
      - 2.4.a InformationRelationshipName (supportQualifier)*
    - 2.5 Definiciones de atributo de información*
      - 2.5.1 Definición y valores legales*
      - 2.5.2 Restricciones*
    - 2.6 Notificaciones comunes*
    - 2.7 Modelo de estado de sistema*
  - 3 Definición de interfaz*
    - 3.1 Diagrama de clase que representa interfaces*
    - 3.2 Normas genéricas*
      - 3.b Interfaz InterfaceName (supportQualifier)*
        - 3.b.a Operación OperationName (supportQualifier)*
        - 3.b.b Notificación NotificationName (supportQualifier)*
      - 3.c Hipótesis*
- B.3 Propiedades y herencia de IOC*
  - B.3.1 Propiedades*
  - B.3.2 Herencia*
  - B.3.3 Importación*

A continuación se presentan las directrices para la especificación de los resultados de la fase de análisis.

La plantilla de análisis se basa en el servicio de información 3GPP [b-3GPP TS 32.151], aumentado para cumplir los requisitos adicionales de la metodología (por ejemplo, correlación).

Para una especificación de interfaz de gestión se utilizarán las cláusulas 2.2 y 2.3 de la plantilla "Análisis" indicada en la cláusula B.2. Para un modelo de información (por ejemplo, un modelo de recursos de red), sólo se utilizará la subcláusula 2.2.

La plantilla de análisis utiliza el tipo información como una característica para describir los atributos IOC y los parámetros operación/notificación. En el Anexo E se definen los tipos información válidos que pueden utilizarse y su semántica.

En el Apéndice II se encuentra un ejemplo de utilización de esta plantilla.

Los términos "servicio de análisis|información" y conjunto de "diseños|soluciones" se emplean para denominar especificaciones equivalentes, aunque con distinto nombre, elaboradas por el UIT-T y 3GPP.

## **B.1 Convenios**

### **B.1.1 Calificativos obligatorios, optativos y condicionales**

En esta subcláusula se define una serie de términos utilizados para calificar la relación entre el servicio de análisis|información, los conjuntos de diseños|soluciones y su repercusión en las aplicaciones prácticas de las interfaces. Los calificativos definidos en esta subcláusula se utilizan para calificar únicamente el comportamiento del agente, lo que se considera suficiente para la especificación de interfaces de gestión.

Las especificaciones de análisis|IS definen atributos IOC, interfaces, operaciones, notificaciones, parámetros de operación y parámetros de notificación. Pueden llevar los siguientes calificativos de soporte/lectura/escritura: M, O, CM, CO, C.

Definición de calificativo M (Obligatorio):

- Utilizado para todos los elementos que se han de soportar.

Definición de calificativo O (Optativo):

- Utilizado para elementos que pueden o no soportarse.

Definición de calificativo CM (Condional-Obligatorio):

- Utilizado para los elementos que son obligatorios bajo determinadas condiciones, en concreto:
  - Todos los elementos que llevan el calificativo de soporte CM llevarán la restricción correspondiente definida en la especificación de recomendación|IS. Si se cumple la restricción especificada, se soportarán los elementos.

Definición de calificativo CO (Condional-Optativo):

- Utilizado para los elementos que son optativos bajo determinadas condiciones, en concreto:
  - Todos los elementos que llevan el calificativo de soporte CO llevarán la restricción correspondiente definida en la especificación de recomendación|IS. Si se cumple la restricción especificada, se podrán soportar los elementos.

Definición de calificativo C (SS-Condional):

- Utilizado para elementos que sólo son aplicables para determinados conjuntos de diseños|soluciones (SS), pero no para todos ellos.

Las especificaciones de diseño|SS definen los equivalentes SS de los atributos IOC, las operaciones, las notificaciones, los parámetros de operación y los parámetros de notificación. Estos equivalentes SS pueden llevar los siguientes calificativos de soporte/lectura/escritura: M, O, CM y CO.

La correspondencia entre los calificativos de los conceptos análisis|IS y los calificativos de los conceptos SS correspondientes se define de la siguiente manera:

- Para los calificativos M, O, CM y CO, cada elemento IS (operación y notificación, parámetros de operaciones de entrada y salida, parámetro de notificaciones de entrada, relación de información y atributo de información) se corresponderá con su(s) equivalente(s) en todos los SS. El/los equivalente(s) correspondido(s) llevará(n) el mismo calificativo que el calificativo IS.



- Para el calificativo C, cada elemento IS se corresponderá con su(s) equivalente(s) en al menos un SS. El/los equivalente(s) correspondido(s) llevará(n) el mismo calificativo de soporte M u O.

En el Cuadro B.1 se define la semántica de los calificativos de los equivalentes, en términos de soporte desde el punto de vista del agente.

**Cuadro B.1 – Semántica de los calificativos utilizados en los conjuntos de diseños|soluciones**

<b>Equivalente SS correspondido</b>	<b>Obligatorio</b>	<b>Optativo</b>	<b>Condicional-Obligatorio (CM)</b>	<b>Condicional-Optativo (CO)</b>
Equivalente de notificación correspondido	El agente generará la notificación	El agente puede o no generarla	El agente generará esta notificación si se satisface la restricción para este elemento	El agente escogerá si generarla o no. Si opta por generarla, se habrá de satisfacer la restricción para esta notificación
Equivalente de operación correspondido	El agente la soportará	El agente puede o no soportar esta operación. Si no la soporta, el agente rechazará la invocación de operación aduciendo el motivo por el que no soporta esta operación. Se devolverán al gestor el rechazo y su motivo	El agente soportará esta operación si se satisface la restricción para este elemento	El agente puede soportar esta operación si se satisface la restricción para este elemento
Parámetro de entrada del equivalente de operación correspondido	El agente lo aceptará y se comportará en función de su valor	El agente puede o no soportar este parámetro de entrada. Si el agente no soporta este parámetro de entrada, y si éste tiene significado (es decir, no lleva semántica no informativa), el agente rechazará la invocación aduciendo el motivo (por el que no soporta el parámetro). Se devolverán al gestor el rechazo y su motivo	El agente lo aceptará y se comportará en función de su valor si se satisface la restricción para este elemento	El agente puede aceptarlo y comportarse en función de su valor si se satisface la restricción para este elemento
Parámetro de entrada del equivalente de notificación correspondido Y parámetro de salida del equivalente de operación correspondido	El agente facilitará este parámetro	El agente puede facilitar este parámetro	El agente facilitará este parámetro si se satisface la restricción para este elemento	El agente puede facilitar este parámetro si se satisface la restricción para este elemento
Equivalente de atributo IOC correspondido	El agente lo soportará	El agente puede soportarlo	El agente soportará este atributo si se satisface la restricción para este elemento	El agente puede soportar este atributo si se satisface la restricción para este elemento

## B.2 Plantilla de análisis

### 1 Conceptos y antecedentes

*En esta cláusula se incluirá una introducción al análisis de especificación de interfaz de gestión.*

#### 1.a SubClauseTitle

*SubClauseTitle es el nombre de la subcláusula.*

*"a" representa un número, empezando por 1 y aumentándose en 1 a cada nueva subcláusula.*

*La utilización de subcláusulas es optativa.*

### 2 Clases de objetos de información

*Esta cláusula se utilizará para todas las especificaciones (tanto para las especificaciones de interfaz de gestión como para las especificaciones de modelo de información únicamente).*

#### 2.1 Entidades de información importadas y etiquetas locales

*En esta subcláusula se identifica una lista de entidades de información (por ejemplo, clase de objeto de información, interfaz, relación de información, atributo de información) que se han definido en otras especificaciones y que se importan al presente documento. Todas las entidades importadas se tratarán como si se hubieran definido localmente en la especificación de que se trate. Se recurre a la información, entre otras cosas, por motivos de herencia. Cada elemento de esta lista es un par (referencia de etiqueta, etiqueta local). La referencia de etiqueta contiene el nombre de la especificación donde se ha definido, el tipo de la entidad de información y su nombre. La etiqueta local de las entidades de información importadas puede entonces utilizarse en toda la especificación, en lugar de la referencia de etiqueta.*

*Esta información se presenta en un cuadro.*

Referencia de etiqueta	Etiqueta local

*Los elementos importados deben proceder de definiciones independientes del protocolo basadas en esta metodología, pero se pueden importar elementos de otras especificaciones, si es necesario, en interés de la migración de especificaciones específicas del protocolo a lo largo del tiempo.*

*En el Anexo F pueden encontrarse directrices sobre la importación de entidades, así como sobre las propiedades IOC y la herencia.*

#### 2.2 Diagrama de clase

##### 2.2.1 Atributos y relaciones

*Este primer conjunto de diagramas representa todas las clases de objeto de información definidas en este IS con todas sus relaciones y todos sus atributos, incluidas las relaciones con las IOC importadas (de haberlas). Estos diagramas contendrán asimismo las cardinalidades de las clases de objeto de información (para las asociaciones, así como para las relaciones de contención) y también pueden contener los nombres de asociación y los nombres de los roles. Se tratará de diagramas de clase conformes con UML (véase asimismo el Anexo C).*

*No se han de repetir en el diagrama las características (relaciones) de las clases de objeto de información importadas. Las clases de objeto de información se definirán utilizando el estereotipo <<InformationObjectClass>>.*

##### 2.2.2 Herencia

*Este segundo conjunto de diagramas representa la jerarquía de herencia de todas las clases de objeto de información definidas en este IS. Estos diagramas no necesitan contener la jerarquía de herencia completa, pero, como mínimo, contendrán las clases de objeto de información progenitoras de todas las clases de objetos de información definidas en este documento. Por defecto, una clase de objeto de información hereda de la clase de objeto de información "superior". Se tratará de diagramas de clase conformes con UML.*

*No se han de repetir en el diagrama las características (atributos, relaciones) de las clases de objeto de información importadas. Las clases de objeto de información se definirán utilizando el estereotipo <<InformationObjectClass>>.*

*NOTA 1 – Algunas relaciones de herencia presentadas en la subcláusula 2.2.2 pueden repetirse en la subcláusula 2.2.1 para mejorar la legibilidad.*

*NOTA 2 – La herencia de interfaz se muestra en la subcláusula 3.1 y no en la presente subcláusula.*

## 2.3 Definiciones de clases de objetos de información

Cada clase de objeto de información se define utilizando la siguiente estructura.

No se mostrarán los elementos heredados (atributos, etc.), pues ya se han definido en las IOC progenitoras y son, por tanto, válidas para todas las subcláusulas.

### 2.3.a InformationObjectClassName

*InformationObjectClassName* es el nombre de la clase de objeto de información.

"a" representa un número, empezando por 1 y aumentándose en 1 a cada nueva definición de IOC.

#### 2.3.a.1 Definición

La subcláusula <Definition> se escribe en lenguaje natural. La subcláusula <Definition> se refiere a la clase de objeto de información misma. No se pueden encontrar en la definición las características relacionadas con las relaciones que esa clase de objeto puede tener con otras clases de objeto. El lector deberá referirse a la definición de las relaciones para encontrar esa información. Se especificará aquí la información relacionada con la herencia.

También se indicará aquí la información sobre correlación con uno o más requisitos soportados por esta IOC, de la siguiente manera:

Referencia	Etiqueta de requisitos	Observaciones

#### 2.3.a.2 Atributos

La subcláusula <Attributes> presenta la lista de atributos, que son las propiedades gestionables de la clase de objeto. Cada elemento es un tuplo (*attributeName*, *supportQualifier*, *readQualifier*, *writeQualifier*):

- *supportQualifier* indica si el atributo es obligatorio (M), optativo (O), condicional-obligatorio (CM), condicional-optativo (CO), SS-condicional (C) o si no se soporta (-). Los valores permitidos son: obligatorio, optativo, condicional o no soportado ("M", "O", "C", o "-", respectivamente).
- *readQualifier* indica si el gestor ha de poder leer el atributo. Los valores posibles son: obligatorio (M), optativo (O), condicional-obligatorio (CM), condicional-optativo (CO), SS-condicional (C) o no soportado (-). Los valores permitidos son: obligatorio (M), opcional (O) y no soportado (-).
- *writeQualifier* indica si el gestor ha de poder escribir el atributo. La semántica de *writeQualifier* es idéntica a la de *supportQualifier* para "M", "O" y "-". Los valores permitidos son: obligatorio (M), optativo (O) y no soportado (-).

Hay una relación de dependencia entre *supportQualifier*, *readQualifier* y *writeQualifier*. *supportQualifier* indica los requisitos para el soporte del atributo. Para cualquier atributo, independientemente del valor de *supportQualifier*, al menos uno de *readQualifier* o *writeQualifier* ha de ser "M". El valor "O" para el *supportQualifier* implica que el atributo es optativo. Sin embargo, los calificativos de lectura y escritura indican cómo se ha de soportar el atributo optativo, en caso de que se soporte.

El *IRPManager* no puede escribir por definición los atributos internos privados o del agente. Por tanto, su *writeQualifier* siempre será "-".

El *readQualifier* y el *writeQualifier* de un atributo soportado, que sea público, pueden no ser ambos "-".

La utilización de "-" en *supportQualifier* se reserva para el soporte documental de atributos definidos por una IOC "Arquetipo" (véase la subcláusula C.3.5). Los atributos con un *supportQualifier* "-" no se aplican en la IOC que realiza un subconjunto de atributos definidos por el "Arquetipo". El *readQualifier* y el *writeQualifier* no tienen relevancia en este caso. Sin embargo, un atributo no soportado no puede leerse ni escribirse. Por este motivo, el *readQualifier* y el *writeQualifier* será "-" para los atributos no soportados.

Para cualquier IOC que utilice uno o más atributos de un "Arquetipo", se utilizará un cuadro aparte para indicar los atributos soportados. No se recurrirá a dicho cuadro si no se soporta ninguno de los atributos del "Arquetipo". Por ejemplo, si una IOC particular tiene atributos definidos (es decir, atributos no definidos por un "Arquetipo") y encapsula atributos de dos "Arquetipos", la totalidad de los atributos de esa IOC estarán contenidos en tres cuadros distintos.

Esta información se facilita en un cuadro.

Nombre del atributo	Calificativo de soporte (Support qualifier)	Calificativo de lectura (Read qualifier)	Calificativo de escritura (Write qualifier)	ID de requisito

### 2.3.a.3 Restricciones de atributo

La subcláusula <Attribute constraints> presenta restricciones entre atributos, que siempre se han de considerar verdaderas. Esas propiedades se considerarán siempre verdaderas durante toda la existencia del atributo y, en concreto, no se han de repetir en las condiciones a priori o a posteriori de las operaciones o notificaciones.

NOTA – Esta subcláusula no ha de incluirse cuando no haya restricciones de atributo que definir.

### 2.3.a.4 Relaciones

En la subcláusula <Relationship> se presenta la lista de relaciones en que participa esta clase. Cada elemento es un relationshipName.

Las relaciones se enumerarán en un cuadro como el siguiente:

Relación	ID de requisito

Cada nombre de relación debe ser una referencia (y preferentemente también un hipervínculo) a la subcláusula correspondiente de la cláusula 2 (clases de objeto de información).

NOTA – Esta subcláusula es optativa y puede no incluirse, pues todas las relaciones están representadas en el diagrama de clase de la subcláusula 2.2.1.

### 2.3.a.5 Diagrama de estados

La subcláusula <State diagram> contiene los diagramas de estados. El diagrama de estados de una clase de objeto de información define los estados permitidos de esta clase de objeto de información y las transiciones entre esos estados. Un estado se expresa en términos de valores de atributo individual o de una combinación de valores de atributo o de participación en las relaciones de la clase de objeto de información que se define. Debe ser un diagrama de estados conforme con UML.

NOTA – No es necesario incluir esta subcláusula cuando no haya diagrama de estados que definir.

### 2.3.a.6 Notificaciones

Para esta IOC la subcláusula <Notifications> presenta:

- optativamente, una referencia a las notificaciones comunes definidas en la subcláusula 2.6 como válidas para esta IOC, y
- optativamente, una lista de las notificaciones que se excluirán de la lista de notificaciones comunes (definidas en la subcláusula 2.6) para esta IOC (téngase en cuenta que no se pueden excluir las notificaciones heredadas de la(s) IOC progenitora(s)),  
y
- optativamente, una lista de notificaciones aplicables a esta IOC, y que pueden o no estar definidas en las notificaciones comunes de la subcláusula 2.6.

Las notificaciones identificadas en esta subcláusula son notificaciones que pueden emitirse por la interfaz de gestión, donde los parámetros "clase de objeto" e "instancia de objeto" del encabezamiento de notificación (véase la Nota 2) de estas notificaciones identifican una instancia de la IOC definida por la subcláusula de nivel superior (es decir, la subcláusula 2.3.a).

Las notificaciones identificadas en esta subcláusula pueden proceder de objetos de aplicación cuyo identificador se corresponde en la aplicación al identificador de instancia de objeto utilizado por la interfaz de gestión. Por tanto, la presencia de notificaciones en esta subcláusula (es decir, la subcláusula 2.3.a.6) no implica que las notificaciones tengan su origen en una instancia de la IOC definida por la subcláusula de nivel superior (es decir, la subcláusula 2.3.a) ni las identifica como tales.

La información relacionada con la opción c) anterior se presenta en un cuadro como el siguiente:

Nombre	Calificativo	ID de requisito	Notas

NOTA 1 – Esta subcláusula y el cuadro pueden omitirse.

NOTA 2 – El encabezamiento de notificación se define en el servicio de información IRP de notificación [b-3GPP TS 32.302].

NOTA 3 – El calificativo de una notificación, especificado en el Cuadro de notificación, indica si esa notificación puede llevar la instancia DN en la notificación. El calificativo de una notificación, especificado en la especificación de gestión, indica el nivel de soporte relativo a la emisión de la notificación en cuestión.

Un gestor puede recibir una notificación-XYZ que lleva una instancia de DN de clase-ABC única y exclusivamente si:

- 1) el Cuadro de notificación de clase-ABC define la notificación-XYZ, y
- 2) la aplicación de la instancia de clase-ABC soporta la notificación-XYZ, y
- 3) una interfaz de gestión define la notificación-XYZ, y
- 4) la aplicación de la interfaz de gestión soporta la notificación-XYZ.

## 2.4 Definiciones de relaciones de información

En esta subcláusula se enumeran todas las relaciones soportadas por esta Recomendación/Especificación en un cuadro como el que sigue. El calificativo de soporte se define igual que para los atributos de la cláusula B.1.

Relación	Calificativo de soporte	ID de requisito

Las relaciones de información se definen de acuerdo con la siguiente estructura.

No se mostrarán las relaciones heredadas, pues ya están definidas por la(s) IOC progenitora(s) y, por tanto, son válidas para todas las subclases.

### 2.4.a InformationRelationshipName (supportQualifier)

InformationRelationshipName es el nombre de la relación de información seguida por un calificativo (véase la cláusula B.1).

"a" representa un número, empezando por 1 y aumentándose en 1 a cada nueva definición de relación de información.

#### 2.4.a.1 Definición

La subcláusula <Definition> se escribe en lenguaje natural.

#### 2.4.a.2 Roles

La subcláusula <Roles> identifica los roles que desempeñan las clases de objeto en la relación. Cada elemento es un par (roleName, roleDefinition).

Esta información se presenta en un cuadro.

Nombre	Definición

#### 2.4.a.3 Restricciones

La subcláusula <Constraints> contiene la lista de propiedades especificando las invariables semánticas que se han de preservar en la relación. Cada elemento es un par (propertyName, propertyDefinition). Esas propiedades siempre se considerarán verdaderas durante toda la existencia de la relación y no es necesario repetirlas en las condiciones a priori o a posteriori de las operaciones o las notificaciones.

Esta información se presenta en un cuadro.

Relación	Calificativo de soporte	ID de requisito

## 2.5 Definiciones de atributo de información

Los atributos de información se definen de acuerdo con la siguiente estructura.

No se mostrarán los atributos heredados, pues ya están definidos en la(s) IOC progenitora(s) y, por tanto, son válidos para todas las subclases.

### 2.5.1 Definición y valores legales

Para cada atributo que se define, en esta subcláusula se indican su nombre de atributo, su definición, escrita en lenguaje natural, un tipo de información (véase el Anexo E) y una lista optativa de valores legales soportados por el atributo.

Cuando los valores legales pueden enumerarse, cada elemento es un par (Legal Value Name, Legal Value Semantics), a menos que la semántica de valor legal se aplique a varios valores, en cuyo caso la semántica se indicará sólo una vez. Cuando los valores legales no puedan enumerarse, la lista de valores legales se definirá con una sola definición.

Esta información se presenta en un cuadro.

Nombre de atributo	Definición	Tipo de información/valores legales

### 2.5.2 Restricciones

La subcláusula <Constraints> indica si hay alguna restricción que afecta a los atributos. Cada restricción se define por un tuplo (propertyName, affected attributes, propertyDefinition). Las PropertyDefinitions se expresan en lenguaje natural.

Esta información se presenta en un cuadro.

Nombre	Atributo(s) afectado(s)	Definición

## 2.6 Notificaciones comunes

La subcláusula <Common Notifications> presenta una lista de notificaciones a las que se puede referir cualquier IOC definida por esta especificación de interfaz de gestión. Estas notificaciones sólo son aplicables a las IOC que se refieren a esta subcláusula en la subcláusula 2.3.a.6.

Esta información se presenta en un cuadro.

Nombre	Calificativo	Notas

NOTA – Esta subcláusula puede omitirse si no hay notificaciones comunes.

## 2.7 Modelo de estado de sistema

Algunas configuraciones de información son lo suficientemente especiales o complejas para justificar la utilización de un diagrama de estados para aclararlas. El diagrama de estados de esta subcláusula define los estados permitidos del sistema y las transiciones entre esos estados. Los estados se expresan en términos de una combinación de restricciones de valores de atributo o de participación en las relaciones de una o más clases de objeto de información.

### 3 Definición de interfaz

Esta cláusula se utilizará para todas las especificaciones de interfaz de gestión y es optativa para las especificaciones de modelo de información únicamente.

#### 3.1 Diagrama de clase que representa interfaces

Cada interfaz se define en el diagrama. Se tratará de un diagrama de clase conforme con UML (véase también el Anexo C).

Las interfaces se definen utilizando un estereotipo <<Interface>>. Cada interfaz contiene un conjunto de operaciones o de notificaciones que son obligatorias; o una única operación o una única notificación optativas. Se utilizan estereotipos (véase el Anexo C) para especificar las interfaces optativas u obligatorias. En el diagrama de clase, cada operación y notificación de una interfaz se calificarán como "públicas" añadiendo el símbolo "+" antes de cada operación o notificación.

NOTA –En esta subcláusula puede mostrarse la herencia de interfaces.

#### 3.2 Normas genéricas

Las siguientes normas atañen a todas las especificaciones. Simplemente se copiarán como parte de la especificación.

Norma 1: Cada operación con al menos un parámetro de entrada soportará una condición a priori `valid_input_parameter`, que indica que todos los parámetros de entrada serán válidos con respecto a su tipo de información. Además, cada una de estas operaciones soportará una excepción `operation_failed_invalid_input_parameter`, creada cuando la condición a priori `valid_input_parameter` es falsa. La excepción tiene el mismo estado de entrada y salida.

Norma 2: Cada operación con al menos un parámetro de entrada optativo soportará un conjunto de condiciones a priori `supported_optional_input_parameter_xxx` donde "xxx" es el nombre del parámetro de entrada optativo y la condición a priori indica que la operación soporta el parámetro de entrada optativo nombrado. Además, cada una de estas operaciones soportará una excepción `operation_failed_unsupported_optional_input_parameter_xxx`, creada cuando (a) la condición a priori `supported_optional_input_parameter_xxx` es falsa y (b) el parámetro de entrada optativo nombrado transporta información. La excepción tiene el mismo estado de entrada y salida.

Norma 3: Cada operación soportará una excepción genérica `operation_failed_internal_problem`, creada cuando hay un problema interno y no se puede completar la operación. La excepción tiene el mismo estado de entrada y salida.

NOTA – Quedan en estudio las consideraciones de seguridad y las normas genéricas resultantes.

#### 3.b Interfaz `InterfaceName (supportQualifier)`

`InterfaceName` es el nombre de la interfaz seguido por un calificativo (véase la cláusula B.1).

"b" representa un número, empezando por 3 y aumentándose en 1 a cada nueva definición de interfaz.

Cada interfaz se define por su nombre y por una secuencia de operaciones o notificaciones, como se define a continuación.

Cada operación se define de acuerdo con la siguiente estructura.

NOTA – Queda en estudio la agrupación de operaciones/segmentación de contenidos de interfaz y la denominación de interfaces.

#### 3.b.a Operación `OperationName (supportQualifier)`

`OperationName` es el nombre de la operación seguido por un calificativo (véase la cláusula B.1).

"a" representa un número, empezando por 1 y aumentándose en 1 a cada nueva definición de operación.

#### 3.b.a.1 Definición

La subcláusula <Definition> se escribe en lenguaje natural.

Se definirá aquí la información sobre la correlación con uno o más requisitos soportados por esta operación, de la siguiente manera:

Referencia	Etiqueta de requisitos	Observaciones

### 3.b.a.2 Parámetros de entrada

Lista de parámetros de entrada de la operación. Cada elemento es un tuplo (Nombre del parámetro, calificativo de soporte, tipo de información (véanse el Anexo E y la Nota a la cláusula E.2) y una lista optativa de valores legales soportados por el parámetro, observaciones). Los valores legales del calificativo de soporte se especifican en la cláusula B.1.

Esta información se presenta en un cuadro.

Nombre del parámetro	Calificativo de soporte	Tipo de información/ valores legales correspondientes	Observaciones

NOTA – El tipo de información califica el parámetro del Nombre del parámetro. Cuando los valores legales pueden enumerarse, cada elemento es un par (Legal Value Name, Legal Value Semantics), a menos que la semántica de valor legal se aplique a varios valores, en cuyo caso la definición se presenta sólo una vez. Cuando los valores legales no pueden enumerarse, la lista de valores legales está definida por una única definición.

### 3.b.a.3 Parámetros de salida

Lista de parámetros de salida de la operación. Cada elemento es un tuplo (nombre del parámetro, calificativo de soporte, información/tipo de información correspondiente (véanse el Anexo E y la Nota a la cláusula E.2) y una lista optativa de valores legales soportados por el parámetro, observaciones). Los valores legales del calificativo de soporte se especifican en la cláusula B.1.

Esta información se presenta en un cuadro.

Nombre del parámetro	Calificativo de soporte	Información/ tipo de información/ valores legales correspondientes	Observaciones

NOTA – El tipo de información califica el parámetro del Nombre del parámetro. Cuando los valores legales pueden enumerarse, cada elemento es un par (Legal Value Name, Legal Value Semantics), a menos que la semántica de valor legal se aplique a varios valores, en cuyo caso la definición se presenta sólo una vez. Cuando los valores legales no pueden enumerarse, la lista de valores legales está definida por una única definición.

Este cuadro también incluye un parámetro especial "estado" para indicar el estado de compleción de la operación (éxito, éxito parcial, motivo de fallo, etc.).

### 3.b.a.4 Condición a priori

Una condición a priori es un conjunto de afirmaciones unidas por los operadores lógicos Y, O y NO (AND, OR y NOT). La condición a priori ha de ser verdadera antes de invocar la operación.

Cada afirmación está definida por un par (propertyName, propertyDefinition). Todas las afirmaciones que constituyen la condición a priori se indican en un cuadro.

Nombre de la afirmación	Definición

### 3.b.a.5 Condición a posteriori

Una condición a posteriori es un conjunto de afirmaciones unidas por los operadores lógicos Y, O y NO (AND, OR y NOT). La condición a posteriori ha de ser verdadera tras la compleción de la operación.

Cuando no se dice nada en una condición a posteriori en una entidad de información, se supone que esa entidad de información no ha cambiado con respecto a lo indicado en la condición a priori.



Cada afirmación está definida por un par (*propertyName*, *propertyDefinition*). Todas las afirmaciones que constituyen la condición a posteriori se indican en un cuadro.

Nombre de la afirmación	Definición

### 3.b.a.6 Excepciones

Lista de excepciones que puede crear la operación. Cada elemento es un tuplo (*exceptionName*, *condition*, *ReturnedInformation*, *exitState*).

#### 3.b.a.6.c *exceptionName*

*ExceptionName* es el nombre de una excepción.

"c" representa un número, empezando por 1 y aumentándose en 1 a cada nueva definición de excepción.

Esta información se presenta en un cuadro.

Nombre de la excepción	Definición	
	Condición	
	Información de vuelta	
	Estado de salida	
	Condición	
	Información de vuelta	
	Estado de salida	

### 3.b.a.7 Restricciones

La subcláusula <Constraints> presenta las restricciones para la operación o sus parámetros.

NOTA – Esta subcláusula puede omitirse cuando no hay restricciones que definir.

#### 3.b.b Notificación *NotificationName* (*supportQualifier*)

*NotificationName* es el nombre de la notificación seguido por un calificativo (véase la cláusula B.1).

"b" representa un número, empezando por 1 y aumentándose en 1 a cada nueva definición de notificación.

##### 3.b.b.1 Definición

La subcláusula <Definition> se escribe en lenguaje natural.

Se debe definir aquí la información sobre la correlación con uno o más requisitos soportados por esta notificación, de la siguiente manera:

Referencia	Etiqueta de requisitos	Observaciones

##### 3.b.b.2 Parámetros de entrada

Lista de parámetros de entrada de la notificación. Cada elemento es un tuplo (nombre del parámetro, calificativos, información/tipo de información correspondiente (véase el Anexo E y la Nota a la cláusula E.2) y una lista optativa de valores legales soportados por el parámetro, observaciones).

La columna "calificativos" contiene los dos calificativos, el calificativo de soporte (véase la cláusula B.1) y el calificativo de filtrado (*Filtering Qualifier*), separados por una coma. El calificativo de filtrado indica si el parámetro de la notificación puede filtrarse o no. Los valores son sí (Y) o no (N).

Esta información se presenta en un cuadro.

Nombre del parámetro	Calificativos	Información/ tipo de información/ valores legales correspondientes	Observaciones

NOTA – El tipo de información califica el parámetro del Nombre del parámetro. Cuando los valores legales pueden enumerarse, cada elemento es un par (Legal Value Name, Legal Value Semantics), a menos que la semántica de valor legal se aplique a varios valores, en cuyo caso la definición se presenta sólo una vez. Cuando los valores legales no pueden enumerarse, la lista de valores legales está definida por una única definición.

### 3.b.b.3 Evento desencadenante

El evento desencadenante de la notificación que se enviará es la transición del estado información definido por la subcláusula "de estado" al estado de información definido por la subcláusula "a estado".

#### 3.b.b.3.1 De estado

Esta subcláusula es un conjunto de afirmaciones unidas por los operadores lógicos Y, O y NO (AND, OR y NOT).

Cada afirmación está definida por un par (propertyName, propertyDefinition). Todas las afirmaciones que constituyen el estado "de estado" se indican en un cuadro.

Nombre de la afirmación	Definición

#### 3.b.b.3.2 A estado

Esta subcláusula es un conjunto de afirmaciones unidas por los operadores lógicos Y, O y NO (AND, OR y NOT). Cuando no se dice nada en el "a estado" en relación con una entidad de información, se supone que esa entidad de información no ha cambiado con respecto a lo que se dice en el "de estado".

Cada afirmación está definida por un par (propertyName, propertyDefinition). Todas las afirmaciones que constituyen el estado "a estado" se indican en un cuadro.

Nombre de la afirmación	Definición

#### 3.b.b.4 Restricciones

La subcláusula <Constraints> presenta las restricciones para la notificación o sus parámetros.

NOTA – Esta subcláusula puede omitirse si no hay restricciones por definir.

### 3.c Hipótesis

Esta subcláusula contiene uno o más diagramas secuenciales que describen, cada uno de ellos, una hipótesis posible. Se tratará de diagramas secuenciales conformes con UML. Esta subcláusula es optativa.

## **B.3 Propiedades y herencia de IOC**

### **B.3.1 Propiedades**

Las propiedades de una IOC (excluida la IOC Soporte) se especifican en los siguientes términos:

- a) Atributo(s) de IOC, incluidas su semántica y sintaxis, sus gamas de valores legales y calificativos de soporte. Los atributos de IOC no se limitan a la gestión de configuración, sino que también incluyen los relacionados con, por ejemplo, 1) la gestión de rendimiento (es decir, tipos de medición), 2) la gestión de rastreo y 3) la gestión de contabilidad.
- b) El comportamiento no específico de un atributo asociado con una IOC (véase la Nota 1).  
NOTA 1 – Por ejemplo, el enlace entre A y B es optativo. Es obligatorio si la instancia A pertenece a una instancia ManagedElement, mientras que la instancia B pertenece a otra instancia ManagedElement. Este comportamiento de enlace es un comportamiento no específico del atributo. Se prevé que este comportamiento, al igual que otros, se herede.
- c) Relación(es) de una IOC con otra(s) IOC.
- d) Tipo(s) de notificación IOC y sus calificativos.
- e) Relación de la IOC con sus progenitores (véase la Nota 2). Hay tres casos mutuamente excluyentes:
  - 1) La IOC es abstracta y no se han designado todavía los progenitores.
  - 2) La IOC es abstracta y se han designado todos los progenitores posibles y, si se puede, se ha designado una subclase IOC como IOC raíz.
  - 3) La IOC no es abstracta y se han designado todos los progenitores posibles y, si se puede, se ha designado una IOC como IOC raíz.

Una instancia IOC es una IOC raíz o tiene única y exclusivamente un progenitor.

NOTA 2 – La relación progenitor-vástago en esta subcláusula es el nombre del progenitor que contiene la relación de vástago.

- f) Relación de la IOC con sus vástagos. Hay tres casos mutuamente excluyentes:
  - 1) Una IOC no tendrá IOC vástagos (relación de contención de nombre).
  - 2) Una IOC puede tener IOC vástagos. Puede especificarse el número máximo de instancias por IOC vástago. Una IOC puede designar que los objetos específicos del fabricante no pueden considerarse IOC vástagos.
  - 3) Una IOC sólo puede tener las IOC vástagos especificadas (o sus subclases). Puede especificarse el número máximo de instancias por IOC vástago. Una IOC puede designar que los objetos específicos del fabricante no pueden considerarse IOC vástagos.
- g) Posibilidad o imposibilidad de instanciación de una IOC (es decir, si la IOC es una IOC abstracta).
- h) Atributo para denominación.

### **B.3.2 Herencia**

Una IOC (subclase) hereda de otra IOC (superclase) de manera que la subclase tendrá todas las propiedades de la superclase.

La subclase puede modificar los calificativos de soporte heredados de optativos a obligatorios, pero no al contrario. La subclase puede modificar los calificativos de soporte heredados de condicional-optativo a condicional-obligatorio, pero no al contrario.

Una IOC puede ser la superclase de muchas IOC. Una subclase no puede tener más de una superclase.

La subclase puede:

- a) Añadir (en comparación con los de su superclase) atributos exclusivos, incluido su comportamiento, gamas de valores legales y calificativos de soporte. Cada atributo adicional tendrá su propio nombre de atributo exclusivo (entre todos los atributos añadidos y heredados).
- b) Añadir comportamientos independientes del atributo por IOC. Estos comportamientos no serán contrarios al comportamiento de superclase heredado.
- c) Añadir relaciones con IOC. Cada relación adicional tendrá un nombre exclusivo (entre todas las relaciones añadidas y heredadas).
- d) Añadir tipos de notificación adicionales y sus calificativos.
- e) Designar todos los progenitores posibles (y sus subclases), si la superclase tiene Property-e-1 de manera que una IOC tendrá Property-e-2 o Property-e-3. Limitar los progenitores posibles (y sus subclases) y/o eliminar la capacidad de la subclase para ser IOC raíz, si la superclase tiene Property-e-2 o Property-e-3.
- f) Añadir IOC vástagos, si la superclase tiene Property-f-2, de manera que una IOC tendrá Property-f-3. Limitar las IOC vástagos permitidas (o sus subclases), si la superclase tiene Property-f-3.
- g) Especificar si una IOC puede o no instanciarse (es decir, la IOC es una IOC abstracta).
- h) Limitar la gama de valores legales de un atributo de superclase que tiene una gama de valores legales.

### **B.3.3 Importación**

Para facilitar la reutilización de definiciones de IOC en las especificaciones IRP, se utiliza un mecanismo de importación mediante el cual una especificación IRP (denominada IRP sujeto) reutiliza la definición de IOC de otra especificación IRP. Cuando la especificación IRP sujeto importa una IOC, no puede modificar las propiedades de la IOC importada. Si se ha de modificar la IOC importada, se debe utilizar la herencia para definir su propia nueva clase.

## Anexo C

### Repertorio UML MISM

(Este anexo forma parte integrante de la presente Recomendación.)

A continuación se presentan las directrices para la especificación de los resultados de la fase análisis, de acuerdo con el repertorio de lenguaje de modelización unificado (UML) 3GPP [b-3GPP TS 32.152].

#### C.1 Introducción

UML ofrece un rico conjunto de conceptos, notaciones y elementos de modelo para modelizar sistemas distribuidos. No es necesario utilizar todas las notaciones y elementos de modelo UML para la especificación del análisis. En este anexo se documentan las notaciones y elementos de modelo UML necesarios y suficientes, incluidos los construidos por el mecanismo de extensión UML <<stereotype>>, que se utilizarán para elaborar especificaciones independientes del protocolo. En conjunto, estas notaciones y elementos de modelo se denominan repertorio de modelización UML.

Las Recomendaciones que sigan la metodología emplearán la notación y los elementos de modelo UML de este repertorio y podrán asimismo utilizar otras notaciones y elementos de modelo UML, según proceda.

#### C.2 Elementos de modelo básicos

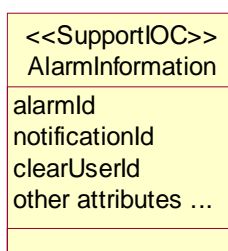
##### C.2.1 Aspectos generales

UML define una serie de elementos de modelo básicos. En esta subcláusula se indica el subconjunto seleccionado para el repertorio. La semántica de los elementos seleccionados se define en [OMG UML].

##### C.2.2 Atributo

Véase la subcláusula 3.25 de [OMG UML].

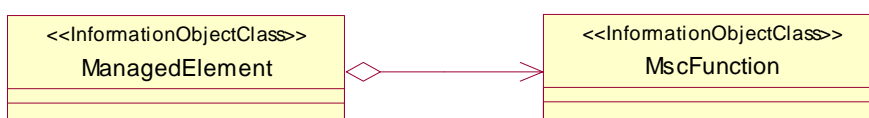
En esta imagen se muestran algunos atributos, representados como cadenas en el compartimento atributo de la clase AlarmInformation.



##### C.2.3 Agregación

Véase la subcláusula 3.43.2.5 de [OMG UML].

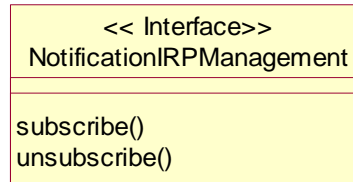
En esta imagen se muestra un rombo vacío ligado al extremo de un trayecto para indicar la agregación. El rombo está ligado a la clase agregada.



### C.2.4 Operación

Véase la subcláusula 3.26 de [OMG UML].

En esta imagen se ven dos operaciones, representadas como cadenas en el compartimento operación de la clase NotificationIRPManagement, que la instancia de NotificationIRPManagement puede solicitar que se realicen. La operación tiene un nombre, por ejemplo, abono (subscribe) y una lista de argumentos (no representada).

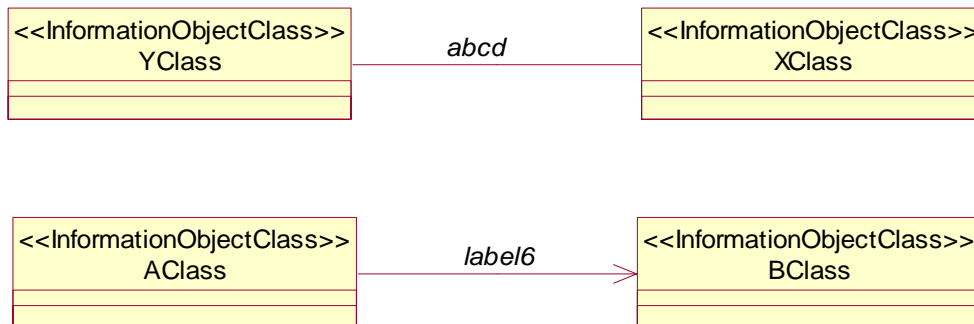


### C.2.5 Asociación y nombre de asociación

Véase la subcláusula 3.41 de [OMG UML].

Estas dos imágenes muestran una asociación binaria entre exactamente dos elementos de modelo. La asociación pueden incluir la posibilidad de relacionar un elemento de modelo consigo mismo. La primera imagen muestra una asociación bidireccional en la que cada elemento de modelo conoce al otro. En la segunda imagen se ve una asociación unidireccional (indicada por una flecha abierta en el extremo de elemento modelo objetivo) en la que sólo el elemento de modelo origen conoce el elemento de modelo objetivo y no al contrario.

La asociación puede recibir un nombre, como *abcd* y *label6* en las imágenes siguientes.



### C.2.6 Relación de realización

Véase la subcláusula 2.5.2.1 de [OMG UML].

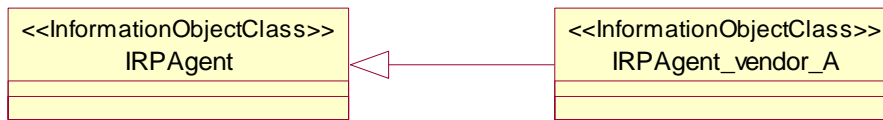
En esta imagen se muestra la relación de realización entre un elemento de modelo AlarmIRPOperations\_1 y otro elemento de modelo, AlarmIRP. Este último (el elemento de modelo objetivo) aplica el primero. El elemento de modelo objetivo debe ser una <<Interface>>.



### C.2.7 Relación de generalización

Véase la subcláusula 3.50 de [OMG UML].

En esta imagen se muestra una relación de generalización entre un elemento más general (el agente) y un elemento más específico (el Agent\_vendedor\_A), plenamente coherente con el primer elemento y que añade información adicional.



### C.2.8 Relación de dependencia

Véase la subcláusula 3.51 de [OMG UML].

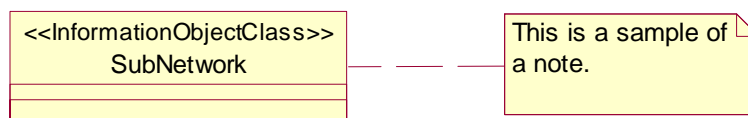
Esta imagen muestra que las instancias BClass tienen una relación semántica con las instancias AClass. Se trata de una situación en que todo cambio del elemento objetivo necesitará un cambio del elemento origen de la dependencia.



### C.2.9 Nota

Véase la subcláusula 3.11 de [OMG UML].

La siguiente imagen muestra una nota como un rectángulo con la esquina superior derecha "doblada". La nota puede contener cualquier texto. Aparece en un diagrama y puede estar anexada a cero o más elementos de modelización mediante una línea discontinua.

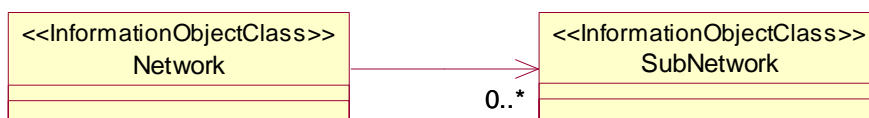


### C.2.10 Multiplicidad (también denominada, cardinalidad)

Véase la subcláusula 3.44 de [OMG UML].

La siguiente imagen muestra una multiplicidad ligada al extremo de un trayecto de asociación. El significado de esta multiplicidad es uno a muchos. La(s) instancia(s) Network está(n) asociada(s) a cero, una o más instancias SubNetwork.

En anteriores versiones de [b-3GPP TS 32.152], la cardinalidad cero puede indicar que IOC tiene la característica denominada "estado transitorio". Por ejemplo, indica que la instancia aún no se ha creado, pero está siéndolo. En esta versión de la metodología, la cardinalidad cero no se utilizará para indicar esta característica, pues se considera que es inherente a todas las IOC. Se considera que todas las IOC definidas tienen la característica "estado transitorio" inherente.



### C.2.11 Nombre de rol

Véase la subcláusula 3.43.2.6 de [OMG UML].

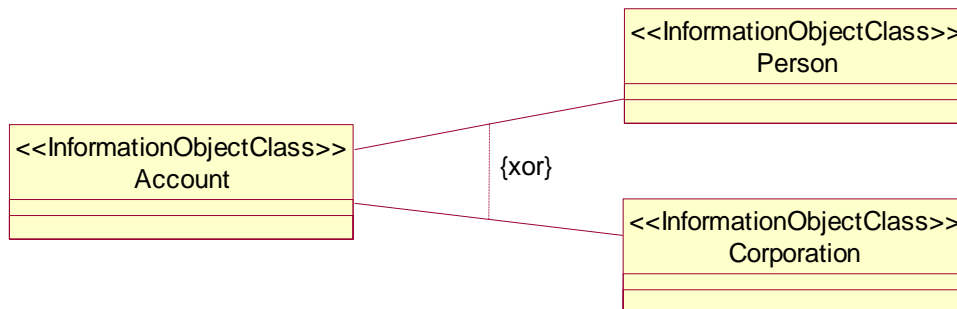
En esta imagen se ve una Person (digamos, la instancia John) asociada a una Company (cuyo DN es, por ejemplo "Company=XYZ"). Navegamos por la asociación utilizando el extremo de la asociación opuesto, de manera que la Person.theCompany de John tendrá el DN, es decir, "Company=XYZ". Los nombres de rol han de ser sustantivos.



### C.2.12 Restricción Xor (o exclusivo)

Véanse las subcláusulas 2.5.2.3 y 3.42.5.1 de [OMG UML].

En esta imagen se muestra una Account (por ejemplo, account 0960) asociada a una Person (por ejemplo, John Smith) o una Corporation (por ejemplo, ABC Inc).



## C.3 Estereotipos

### C.3.1 Aspectos generales

En esta subcláusula se enumeran todos los estereotipos permitidos que se utilizarán en las especificaciones de interfaces de gestión. El estereotipo <<Interface>> está definido en [OMG UML], y se incluye en esta Recomendación para facilitar la referencia y procurar que sea lo más completa posible. En esta Recomendación se definen otros estereotipos.

**Cuadro C.3-1 – Estereotipos de entidad**

Estereotipo	Clase base	Elementos del metamodelo afectados
Interface	Clase	
ProxyClass	Clase	
Notification	Clase	
Archetype	Clasificador (subcláusula 2.5.2.10 de [OMG UML])	
InformationObjectClass	Clasificador	
SupportIOC	Clasificador	
use	Asociación	
may use	Asociación	
may realize	Asociación	
names	Composición	

### C.3.2 <<Interface>>

*Subcláusula 2.5.2.25 de [OMG UML]:*

"Una interfaz es un conjunto nombrado de operaciones que caracterizan el comportamiento de un elemento. En el metamodelo, una Interface (interfaz) contiene un conjunto de Operations (operaciones) que juntas definen un servicio ofrecido por un Classifier (clasificador) que realiza la Interface. Un Classifier puede ofrecer varios servicios, lo que significa que puede realizar varias Interfaces, y varios Classifiers pueden realizar la misma Interface.



Las Interfaces [pueden o] no pueden tener Attributes (atributos), Associations (asociaciones) o Methods (métodos). Una Interface puede participar en una Association siempre que la Interface no pueda ver la Association. Es decir, que un Classifier (distinto de una Interface) puede tener una Association con una Interface por la que se puede navegar desde el Classifier, pero no desde la Interface."

De la subcláusula 2.5.4.6 de [OMG UML]:

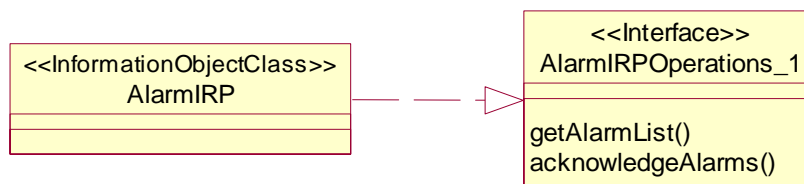
"El objetivo de una interfaz es recopilar una serie de operaciones que constituyen un servicio coherente ofrecido por los clasificadores. Las interfaces ofrecen un medio de dividir y caracterizar los grupos de operaciones. Una interfaz es únicamente un conjunto de operaciones con un nombre. No puede instanciarse directamente."

De la subcláusula 2.5.4.6 de [OMG UML]:

"Varios clasificadores pueden realizar la misma interfaz. Todos ellos deben contener, como mínimo, las operaciones correspondientes a las contenidas en la interfaz. La especificación de una operación contiene la firma de la operación (es decir, su nombre, los tipos de parámetros y el tipo devuelto). Una interfaz no implica ningún tipo de estructura interna del clasificador realizante. Por ejemplo, no incluye qué algoritmo utilizar para realizar una operación. Sin embargo, una operación puede incluir una especificación de los efectos [por ejemplo, con condiciones *a priori* y *a posteriori*] que tendrá su invocación."

### C.3.2.1 Ejemplo

La siguiente imagen muestra un <<Interface>> AlarmIRPOperations\_1 que tiene dos operaciones. Los parámetros de entrada y salida de las operaciones están ocultos (es decir, que no se muestran). La AlarmIRP tiene una relación de realización obligatoria unidireccional con el <<Interface>>.



### Notación de <<Interface>>

## C.3.3 <<ProxyClass>>

### C.3.3.1 Aspectos generales

Esto representa un número de <<InformationObjectClass>>. Encapsula los atributos, enlaces, métodos (u operaciones) e interacciones presentes en el <<InformationObjectClass>> representado.

La semántica de una <<ProxyClass>> es que todos los comportamientos del <<ProxyClass>> estén presentes en el <<InformationObjectClass>> representado. Dado que esta clase es simplemente un representación de otras clases, no puede definir su propio comportamiento, aparte de los ya definidos por el <<InformationObjectClass>> representado.

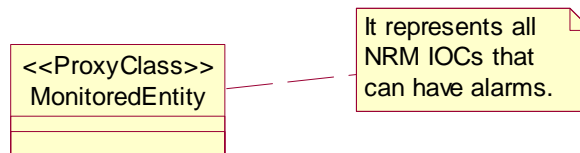
Un <<InformationObjectClass>> concreto puede estar representado por cero, uno o más <<ProxyClass>> o <<Archetype>>. Por ejemplo, el <<InformationObjectClass>> ManagedElement puede tener el <<ProxyClass>> MonitoredEntity y el <<ProxyClass>> ManagedEntity.

La entidad de origen, que tiene una asociación con <<ProxyClass>>, puede acceder a los atributos de <<ProxyClass>>.

### C.3.3.2 Ejemplo

A continuación se muestra un <<ProxyClass>> denominado MonitoredEntity, que representa todos los <<InformationObjectClass>> NRM (por ejemplo, <<InformationObjectClass>> GgsnFunction), en cuyas instancias se supervisan las alarmas.

Téngase en cuenta que <<MonitoredEntity>> no define atributo alguno. Los atributos ya están definidos por todos los <<InformationObjectClass>> representados por <<MonitoredEntity>>.



#### Notación de <<ProxyClass>>

Véanse en el Apéndice V más muestras de la utilización de <<ProxyClass>>.

### C.3.4 <<Archetype>>

#### C.3.4.1 Aspectos generales

Representa un número de propiedades de clase común (por ejemplo, atributos, enlaces, operaciones e interacciones) típicas del <<InformationObjectClass>> representado.

La semántica de <<Archetype>> es que todos los atributos, operaciones de enlace e interacciones encapsulados por <<Archetype>> pueden estar o no presentes en el <<InformationObjectClass>> representado. El <<Archetype>> representa una clase continente de lo más útil en los modelos de análisis independientes de la tecnología, que deberán especificarse más detalladamente y/o tener una correspondencia dentro de un modelo de construcción más completo.

#### C.3.4.2 Ejemplo

A continuación se muestra un <<Archetype>> denominado StateManagement. También se muestra un agente <<InformationObjectClass>> que depende de ese StateManagement. Téngase en cuenta que StateManagement ha definido una serie de atributos (que no se muestran en el diagrama UML). Las clases que dependen de este StateManagement pueden o no utilizar todos los atributos de StateManagement. Dicho de otro modo, al menos uno de los atributos de StateManagement está presente en el agente. El conjunto preciso de atributos de StateManagement utilizados por el agente se especifica en la especificación del agente.



#### Notación de <<Archetype>>

### C.3.5 <<InformationObjectClass>>

#### C.3.5.1 Aspectos generales

Representa una IOC. Cada <<InformationObjectClass>> representa un conjunto de instancias con estructura, comportamiento y relaciones semejantes.

Este <<InformationObjectClass>> y otras clases de información, como <<Interface>>, se corresponden con elementos del modelo específico de la tecnología, como la Clase objeto gestionado GDMO para la tecnología CMIP. La correspondencia entre conceptos de modelización independiente del protocolo y conceptos de modelización específicos de las tecnologías se recoge en las especificaciones específicas del protocolo correspondientes.

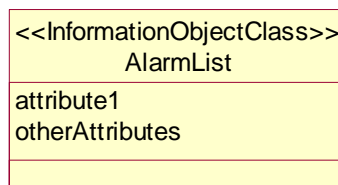
El alcance del nombre de un <<InformationObjectClass>> se limita a la Recomendación en que está especificado, y ese nombre ha de ser exclusivo entre todos los nombres de <<InformationObjectClass>> de esa Recomendación. El nombre de Recomendación recibe la misma consideración que el nombre de lote UML.

<<InformationObjectClass>> es idéntico a la clase UML, excepto que no incluye/define métodos u operaciones.

*Subcláusula 3.22.1 de [OMG UML]:* "Una clase representa un concepto dentro del sistema que se modeliza. Las clases tienen una estructura de datos, un comportamiento y relaciones con otros elementos."

#### C.3.5.2 Ejemplo

A continuación se muestra el <<InformationObjectClass>> AlarmList.



#### Notación de <<InformationObjectClass>>

#### C.3.6 <<use>> y <<may use>>

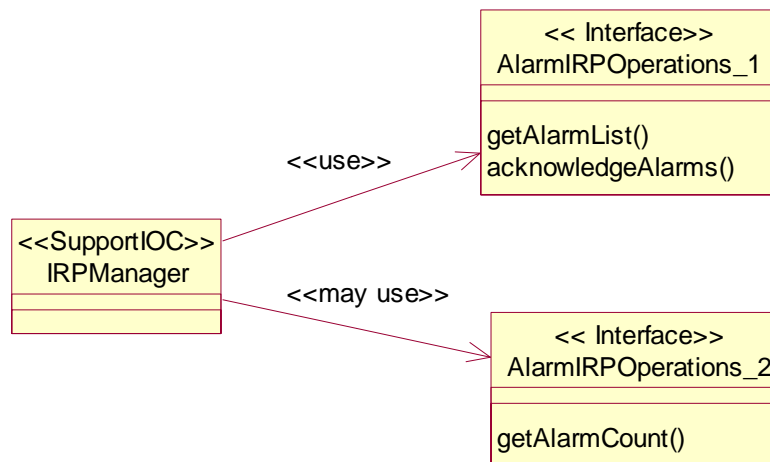
Los arquetipos <<use>> y <<may use>> son asociaciones unidireccionales. El objetivo ha de ser un <<Interface>> o un <<Notification>>.

Cuando el objetivo es <<Interface>>, <<use>> indica que la clase de origen debe tener la capacidad de utilizar el objetivo <<Interface>> de manera que pueda invocar las operaciones definidas por <<Interface>>. Es obligatorio que la entidad de origen soporte la capacidad. <<may use>> indica que la clase de origen puede tener la capacidad de utilizar el objetivo <<Interface>> de manera que pueda invocar las operaciones definidas por <<Interface>>. El soporte de la capacidad por parte de la entidad de origen es optativo.

Cuando el objetivo es <<Notification>>, <<use>> indica que la clase de origen debe ser el emisor de las notificaciones definidas por el objetivo <<Notification>>. Es obligatorio que la entidad de origen soporte la capacidad. <<may use>> indica que la clase de origen puede ser el emisor de las notificaciones definidas por el objetivo <<Notification>>. El soporte de la capacidad por parte de la entidad de origen es optativo.

### C.3.6.1 Ejemplo con el objetivo <<Interface>>

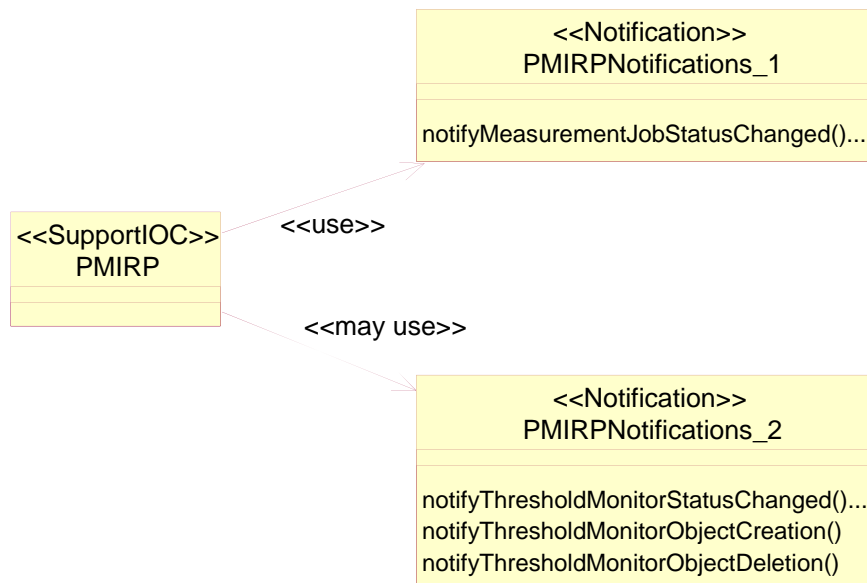
A continuación se ve que el IRPManager utilizará las operaciones definidas por AlarmIRPOperations\_1 y puede utilizar las operaciones definidas por AlarmIRPOperations\_2.



#### Notación de <<use>> y <<may use>> con el objetivo <<Interface>>

### C.3.6.2 Ejemplo con el objetivo <<Notification>>

A continuación se ve que el PMIRP tendrá la capacidad de emitir u originar notificaciones definidas por PMIRPNotifications\_1 y puede tener la capacidad de emitir u originar notificaciones definidas por PMIRPNotifications\_2.



#### Notación de <<use>> y <<may use>> con el objetivo <<Notification>>

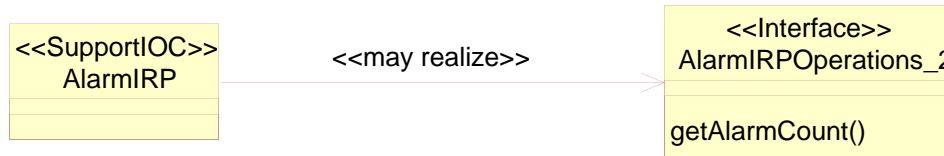
### C.3.7 <<may realize>>

El arquetipo <<may realize>> es una asociación unidireccional. El objetivo debe ser un <<Interface>>. <<may realize>> indica que la entidad de origen puede realizar las operaciones definidas por el objetivo <<Interface>>.

Téngase en cuenta que el elemento básico UML ha definido la asociación de realización (y, por consiguiente, no es necesario definir un estereotipo para tal asociación). La asociación de realización indica que la entidad de origen debe realizar (o aplicar) las operaciones definidas por el objetivo <<Interface>>.

### C.3.7.1 Ejemplo

A continuación se muestra que el AlarmIRP puede realizar la operación de AlarmIRPOperations\_2.



#### Notación de <<may realize>>

### C.3.8 <<names>>

Especifica una composición unidireccional. La instancia objetivo se identifica inequívocamente, dentro del espacio nombre de la entidad de origen, entre todas las demás instancias objetivo del mismo clasificador objetivo y entre las demás instancias objetivo de otros clasificadores que tienen la misma composición <<names>> en el origen.

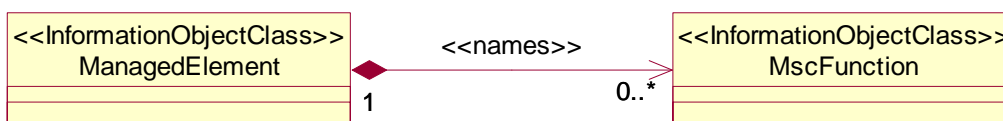
El clasificador de origen y el clasificador de objetivo tendrán ambos un atributo de denominación.

La composición utilizada como acto de contención de nombre ofrece una semántica a la relación todo-parte entre el dominio y los elementos denominados contenidos, incluso aunque sólo sea por el nombre. Desde el punto de vista de la gestión, el acceso a la parte se realiza a través del todo. La multiplicidad se indicará en ambos extremos de la relación.

Una instancia objetivo no puede tener múltiples <<names>> con múltiples orígenes, es decir, que una instancia objetivo no puede participar en múltiples espacios nombre ni pertenecer a ellos.

### C.3.8.1 Ejemplo

A continuación se muestra que todas las instancias de MscFunction se identifican inequívocamente dentro del espacio nombre de la instancia ManagedElement.



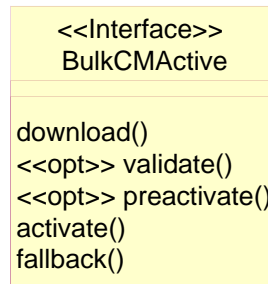
#### Notación de <<names>>

### C.3.9 <<opt>>

El arquetipo <<opt>> (también <<optional>>) permite indicar la calidad optativa de los atributos, parámetros y operaciones (respectivamente) en los diagramas UML.

En ausencia del estereotipo, los atributos, parámetros u operaciones en cuestión son obligatorios.

### C.3.9.1 Ejemplo



#### Notación de <<opt>> para operaciones

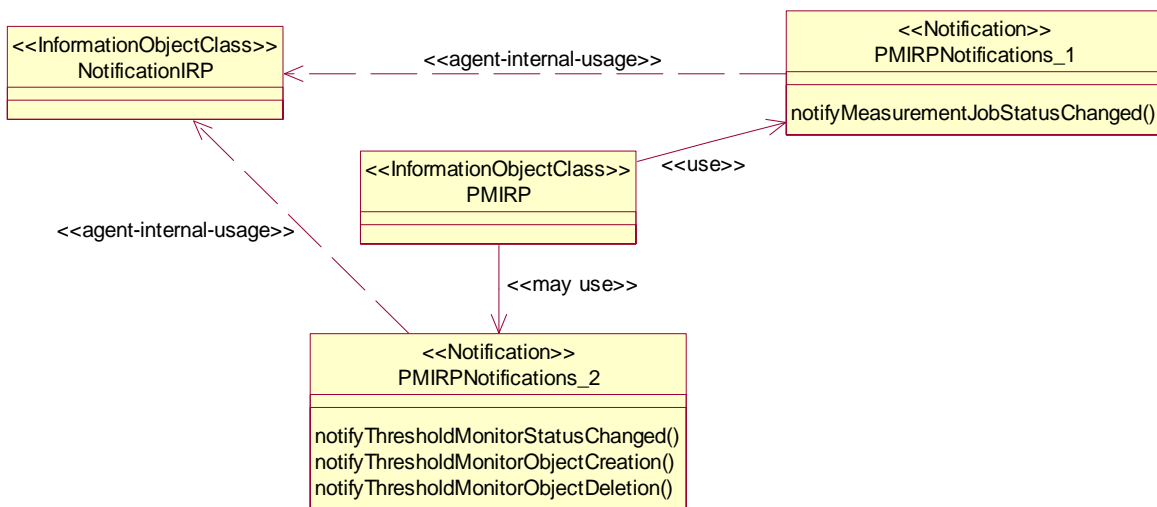
### C.3.10 <<Notification>>

#### C.3.10.1 Aspectos generales

<<Notification>> es un conjunto denominado de notificaciones.

#### C.3.10.2 Ejemplo

A continuación se muestra un <<Notification>> denominado "PMIRPNotifications\_1" y otro <<Notification>> denominado "PMIRPNotifications\_2". Ambos tienen notificaciones. Un ejemplo de notificación puede ser notifyMeasurementJobStatusChanged().



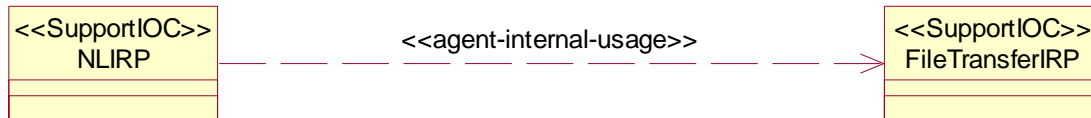
#### Notación de <<Notification>>

### C.3.11 <<agent-internal-usage>>

Se trata de una asociación unidireccional. El origen transmite la información de gestión de red al objetivo. El origen y el objetivo son entidades o procesos que se ejecutan en instancias IRP diferentes, como AlarmIRP, PMIRP. El nombre de las instancias puede estar contenido en el mismo IRPAgent o en instancias IRPAgent diferentes. La información de gestión exacta que se transmite y el mecanismo de transferencia de la información no están normalizados y dependen del fabricante.

### C.3.11.1 Ejemplo

A continuación se muestra que NLIRP (NotificationLog IRP) puede transmitir información de gestión de red a FTIRP (FileTransferIRP).



#### Notación de <<agent-internal-usage>>

### C.3.12 <<SupportIOC>>

Es el descriptor de un conjunto de capacidades de gestión.

<<SupportIOC>> es idéntico a la *clase* UML, excepto que no incluye/define métodos u operaciones.

*Subcláusula 3.22.1 de [OMG UML]:* "Una *clase* representa un concepto en el sistema que se modeliza. Las clases tienen estructura de datos, comportamiento y relaciones con otros elementos."

#### C.3.12.1 Ejemplo

A continuación se muestra un <<SupportIOC>> AlarmList.



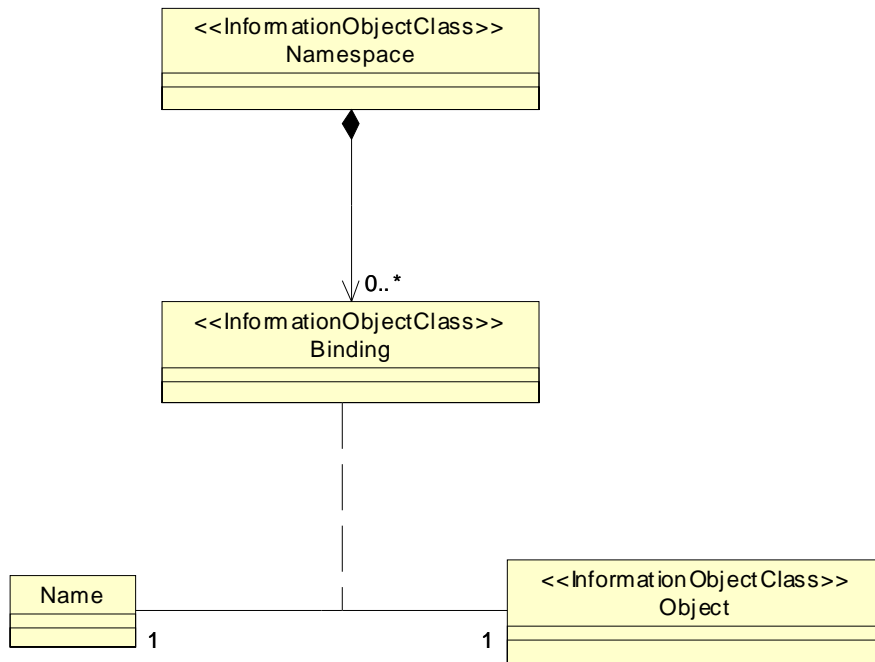
#### Notación de <<SupportIOC>>

## C.4 Clases de asociación

En la subcláusula 3.46 de [OMG UML] se define la clase de asociación de la siguiente manera:

"Una clase de asociación es una asociación que también tiene propiedades de clase (o una clase que tiene propiedades de asociación). Aunque se represente gráficamente como una asociación y una clase, en realidad es sólo un elemento de modelo."

Conviene utilizar clases de asociación cuando una "InformationObjectClass" necesita mantener asociaciones con otras varias "InformationObjectClass" y hay relaciones entre los miembros de las asociaciones dentro del alcance de la "InformationObjectClass" "continente". Por ejemplo, un espacio nombre mantiene una serie de vínculos, uno de ellos entre un nombre y un objeto. Se puede modelizar una "IOC" vinculante como una clase de asociación que ofrece la semántica de vinculación a la relación entre un nombre y cualquier otra "InformationObjectClass". A continuación se muestra la representación gráfica de lo anterior (a título de ejemplo únicamente, no procede de otra Recomendación).



### Ejemplo de clase de asociación

#### C.5 Clase abstracta

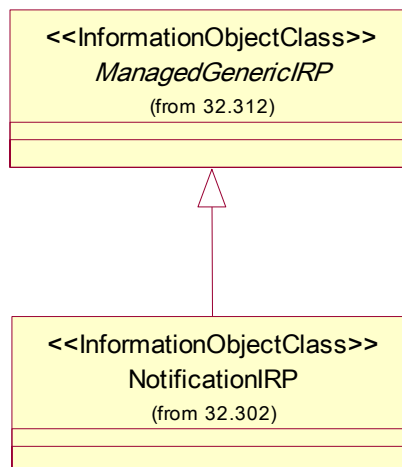
##### C.5.1 Aspectos generales

Especifica una <<InformationObjectClass>> como clase base que heredarán las subclases. Una clase abstracta no puede instanciarse.

La notación de la clase abstracta consiste en la utilización de la cursiva en el nombre de clase de la <<InformationObjectClass>> correspondiente en el diagrama.

##### C.5.2 Muestra

A continuación se muestra que ManagedGenericIRP es una <<InformationObjectClass>> abstracta.



### Notación de clase abstracta



## C.6 Aplicación de <<InformationObjectClass>> y <SupportIOC>>

<<InformationObjectClass>> y <<SupportIOC>> son estereotipos. Estos dos estereotipos tienen un fin semejantes en el sentido de que ambos son un conjunto nombrado de propiedades de gestión. Sin embargo, en el contexto de la gestión de soporte por una interfaz de gestión, su aplicación puede ser diferente. En esta cláusula se resaltan las semejanzas y diferencias de tal aplicación.

	<<InformationObjectClass>>	<<SupportIOC>>
¿Puede ser una clase abstracta?	Sí	Sí
¿Puede ser una clase concreta?	Sí	Sí
¿Puede heredar de <<InformationObjectClass>>?	Sí	No
¿Puede heredar de <<SupportIOC>>?	No	Sí
¿Puede <<InformationObjectClass>> contener el nombre?	Sí	Sí
¿Puede <<SupportIOC>> contener el nombre?	No	Sí
¿Puede la instancia tener un DN?	<<InformationObjectClass>> debe ser una clase de un árbol de denominación, lo que significa que todas sus instancias deben tener un DN.	El autor de la especificación puede utilizar <<SupportIOC>> como clase en un árbol de denominación. En tal caso, todas sus instancias tendrán un DN.
¿Puede un gestor recibir información mediante notificaciones cuyos parámetros objectClass y objectInstance llevan el DN de la instancia?	Sí. Los tipos de notificación emitidos se muestran en el Cuadro de notificación asociado a la definición de clase.	Sí, si <<SupportIOC>> es una clase de un árbol de denominación. Los tipos de notificación emitidos se muestran en el Cuadro de notificación asociado a la definición de clase. No, si <<SupportIOC>> no es una clase de un árbol de denominación.

## **Anexo D**

### **Diseño**

(Este anexo forma parte integrante de la presente Recomendación.)

En este anexo se presentan las directrices para la especificación de diseños específicos del protocolo. Queda en estudio.

## Anexo E

### Definición de tipos de información – tipo repertorio

(Este anexo forma parte integrante de la presente Recomendación.)

En este anexo se define un repertorio de tipos que se utilizará para especificar los tipos de información del modelo conceptual (modelo de análisis/servicio de información).

El repertorio se define como un subconjunto de tipos definidos por ASN.1 [UIT-T X.680] combinados con tipos derivados de los tipos definidos por ASN.1 (cláusula E.4).

En el Cuadro E.1 se recopilan las palabras clave que se utilizarán con cada tipo.

#### E.1 Tipos básicos

Los tipos básicos son tipos que pueden utilizarse directamente para definir atributos y parámetros. Los tipos básicos también pueden emplearse para construir tipos complejos. Los tipos básicos incluyen los siguientes tipos ASN.1:

**E.1.1 tipo entero** cláusula 19 de [UIT-T X.680]

**E.1.2 tipo real** cláusula 21 de [UIT-T X.680]

**E.1.3 tipo booleano** cláusula 18 de [UIT-T X.680]

**E.1.4 tipo cadena de bits** cláusula 22 de [UIT-T X.680]

**E.1.5 tipo nulo** cláusula 24 de [UIT-T X.680]

**E.1.6 tipo tiempo generalizado** cláusula 38 de [UIT-T X.680]

#### E.2 Tipo enumerado

El tipo enumerado de la cláusula 20 de [UIT-T X.680] representa valores enumerados. Todos los valores que puedan utilizar un atributo o un parámetro específicos se incluirán en las columnas de valores legales. Sólo el estilo de nombres enumerados podrá aplicarse al modelo conceptual, es decir, que se deja la identificación de valores concretos (números o cadenas) a los modelos de diseño concretos.

NOTA – Si el número de estos valores es superior a 50, se recomienda definirlos en un apéndice o documento independiente.

#### E.3 Tipos complejos

Los tipos complejos pueden definirse utilizando los siguientes conceptos:

**E.3.1 tipos secuencia** cláusula 25 de [UIT-T X.680]

**E.3.2 tipos opción** cláusula 29 de [UIT-T X.680]

**E.3.3 tipos conjunto** cláusula 27 de [UIT-T X.680]

Además, las listas y conjuntos de tipos complejos se soportan gracias a:

**E.3.4 Tipos secuencia de** cláusula 26 de [UIT-T X.680]

**E.3.5 tipos conjunto de** cláusula 28 de [UIT-T X.680]

## E.4 Tipos útiles

### E.4.1 Tipo cadena

Una cadena representa una cadena de caracteres sin restricción del conjunto de caracteres, es decir:

String ::= UnrestrictedCharacterStringType cláusula 44 de [UIT-T X.680]

### E.4.2 Tipo nombre

Nombre representa el nombre exclusivo de una instancia objeto en el espacio nombre. Puede incluir información de jerarquía de árbol de contención de nombre, pero depende de la aplicación y queda fuera del alcance de esta Recomendación. Formalmente, el tipo nombre se define de siguiente manera:

Name ::= TYPE-IDENTIFIER Anexo A de [UIT-T X.681]

## E.5 Palabras clave

En el Cuadro E.1 se define una lista de palabras clave que se utilizarán en la plantilla de análisis (véase el Anexo B) para definir el tipo de información, por ejemplo:

Nombre del parámetro	Calificativo de soporte	Tipo de información/ valores legales	Observaciones
...			
eventIdList	M	SET OF INTEGER/-	Lista de alarmas que se han de reconocer.

**Cuadro E.1 – Palabras clave**

Tipo	Palabra clave
Tipo entero	INTEGER
Tipo real	REAL
Tipo booleano	BOOLEAN
Tipo cadena de bits	BIT STRING
Tipo nulo	NULL
Tipo tiempo generalizado	GeneralizedTime
Tipo enumerado	ENUMERATED
Tipo secuencia	SEQUENCE
Tipo opción	CHOICE
Tipo conjunto	SET
Tipo secuencia de	SEQUENCE OF
Tipo conjunto de	SET OF
Tipo cadena	String
Tipo nombre	Name

## Anexo F

### Directrices sobre importación de propiedades, herencia y entidad de IOC

(Este anexo forma parte integrante de la presente Recomendación.)

Las siguientes directrices se basan en [b-3GPP TS 32.150].

#### F.1 Propiedad de IOC

Las propiedades de una IOC (incluida la IOC Soporte) se especifican en los siguientes términos:

- a) Atributo(s) de IOC, incluidas su semántica y sintaxis, sus gamas de valores legales y calificativos de soporte. Los atributos de IOC no se limitan a la gestión de configuración, sino que también incluyen los relacionados con, por ejemplo, 1) la gestión de rendimiento (es decir, tipos de medición), 2) las gestión de rastreo y 3) la gestión de contabilidad.
- b) Comportamiento no específico de un atributo asociado con una IOC.

NOTA 1 – Por ejemplo, el enlace entre MscServerFunction y CsMgwFunction es optativo. Es obligatorio si la instancia MscServerFunction pertenece a una instancia ManagedElement mientras que la instancia CsMgwFunction pertenece a otra instancia ManagedElement. Este comportamiento de enlace es un comportamiento no específico del atributo. Se prevé que este comportamiento, al igual que otros, se herede.

- c) Relación(es) de una IOC con otra(s) IOC.
- d) Tipo(s) de notificación IOC y sus calificativos.
- e) Relación de la IOC con sus progenitores (véase la Nota 2). Hay tres casos mutuamente excluyentes:
  - 1) La IOC puede tener cualquier progenitor. En el diagrama UML, la clase tiene el progenitor Any (cualquiera).
  - 2) La IOC es abstracta y se han designado todos los progenitores posibles y, si se puede, se ha designado una subclase IOC como IOC raíz. En el diagrama UML, la clase tiene cero o más progenitores posibles de clases específicas (excepto Any).
  - 3) La IOC es concreta y se han designado todos los progenitores posibles y, si se puede, se ha designado una IOC como IOC raíz. En el diagrama UML, la clase tiene uno o más progenitores posibles de clases específicas (excepto Any).

Una instancia IOC es una IOC raíz o tiene un único y exclusivo progenitor. Sólo 3GPP SA5 puede designar una clase IOC como posible IOC raíz. En la actualidad, sólo las IOC SubNetwork, ManagedElement o MeContext pueden ser IOC raíz.

NOTA 2 – La relación progenitor-vástago en esta subcláusula es el nombre del progenitor que contiene la relación de vástago.

- f) Relación de la IOC con sus vástagos. Hay tres casos mutuamente excluyentes:
  - 1) Una IOC no tendrá IOC vástagos (relación de contención de nombre). En el diagrama UML, la clase no tendrá vástagos.
  - 2) Una IOC puede tener IOC vástagos. Puede especificarse el número máximo de instancias por IOC vástago. Una IOC puede designar que los objetos específicos del fabricante no pueden considerarse IOC vástagos. En el diagrama UML, la clase tendrá el vástago Any.

- 3) Una IOC sólo puede tener las IOC vástagos especificadas (o sus subclases). Puede especificarse el número máximo de instancias por IOC vástago. Una IOC puede designar que los objetos específicos del fabricante no pueden considerarse IOC vástagos. En el diagrama UML, la clase tendrá uno o más vástagos de clases específicas (excepto Any).
- g) Posibilidad o imposibilidad de instanciación de una IOC (es decir, si la IOC es una IOC abstracta).
- h) Atributo para denominación.

## **F.2 Herencia**

Una IOC (subclase) hereda de otra IOC (superclase) de manera que la subclase tendrá todas las propiedades de la superclase.

La subclase puede modificar los calificativos de soporte heredados de optativos a obligatorios, pero no al contrario. La subclase puede modificar los calificativos de soporte heredados de condicional-optativo a condicional-obligatorio, pero no al contrario.

Una IOC puede ser la superclase de muchas IOC. Una subclase no puede tener más de una superclase.

La subclase puede:

- a) Añadir (en comparación con los de su superclase) atributos exclusivos, incluido su comportamiento, gamas de valores legales y calificativos de soporte. Cada atributo adicional tendrá su propio nombre exclusivo (entre todos los atributos añadidos y heredados).
- b) Añadir comportamientos independientes del atributo por IOC. Estos comportamientos no serán contrarios al comportamiento de superclase heredado.
- c) Añadir relaciones con IOC. Cada relación adicional tendrá un nombre exclusivo (entre todas las relaciones añadidas y heredadas).
- d) Añadir tipos de notificación adicionales y sus calificativos.
- e) Designar todos los progenitores posibles (y sus subclases), si la superclase tiene Property-e-1 de manera que una IOC tendrá Property-e-2 o Property-e-3. Limitar los progenitores posibles (y sus subclases) y/o eliminar la capacidad de la subclase para ser IOC raíz, si la superclase tiene Property-e-2 o Property-e-3.
- f) Añadir IOC vástagos, si la superclase tiene Property-f-2 de manera que una IOC tendrá Property-f-3. Limitar las IOC vástagos (o sus subclases), si la superclase tiene Property-f-3.
- g) Especificar si una IOC puede o no instanciarse (es decir, la IOC es una IOC abstracta).
- h) Limitar la gama de valores legales de un atributo de superclase que tiene una gama de valores legales.

## **F.3 Importación de entidad (interfaz, IOC y atributo)**

Las especificaciones de interfaces de gestión definen entidades (por ejemplo, IOC, interfaces y atributos). Para facilitar la reutilización de definiciones de entidades en las especificaciones de interfaces, se utiliza un mecanismo de importación. Cuando una especificación de interfaz de gestión (la especificación sujeto) importa una entidad definida en otra especificación de interfaz de gestión, se considera que la especificación sujeto ha definido la entidad importada en su especificación. Además, la especificación sujeto no puede modificar las propiedades de la entidad importada. Si se necesita una entidad no idéntica, sino semejante a la identidad importada, se debe definir una nueva entidad que herede la entidad importada e introducir cambios en la definición de la nueva entidad.

# Apéndice I

## Ejemplo de requisitos

(Este apéndice no forma parte integrante de la presente Recomendación.)

NOTA – El siguiente ejemplo se basa en la gestión de alarmas, pero se utiliza sólo con fines ilustrativos y no pretende ser un conjunto completo o correcto de requisitos para la gestión de alarmas.

### **1 Conceptos y antecedentes**

Toda evaluación de los elementos de red (NE) y del estado de salud de la red en general necesita que se detecten los fallos en la red y, por consiguiente, se notifiquen las alarmas al sistema operativo (OS) (EM y/o NM).

### **2 Requisitos comerciales**

#### **2.1 Requisitos**

Los fallos que pueden ocurrir en la red pueden agruparse en una de las siguientes categorías:

- Fallos de hardware, es decir, funcionamiento erróneo de alguno de los recursos físicos del NE.
- Problemas de software, por ejemplo, errores de software, incoherencias en las bases de datos.

##### **2.1.1 Detección de fallos**

REQ-FM-FUN-01 La mayoría de los fallos deben tener condiciones bien definidas para poder declarar su presencia o ausencia, es decir, condiciones de existencia de fallos y condiciones de solución de fallos. Todo incidente de este tipo se denominará en este apéndice fallo ADAC. Las entidades de red deben poder reconocer la ausencia de un fallo ADAC previamente detectado, es decir, la solución del fallo, utilizando técnicas similares a las que emplean para detectar la existencia de un fallo.

##### **2.1.2 Anulación de alarmas**

Las alarmas originadas como consecuencia de fallos se han de anular. Para anular una alarma suele ser necesario reparar el fallo correspondiente.

...

REQ-FM-FUN-02 Cada vez que se anula una alarma, el Agente generará un evento de anulación de alarmas adecuado. La anulación de alarmas se define como una alarma.

##### **2.1.3 Reenvío y filtrado de alarmas**

REQ-FM-FUN-03 Para cada fallo detectado, la entidad de red en fallo generará las alarmas (o notificaciones de fallo) correspondientes.

...

### **2.2 Roles de los actores**

Sistema gestionado La entidad ejerce el rol de agente.

Sistema de gestión La entidad ejerce el rol de gestor.

### **2.3 Recursos de telecomunicaciones**

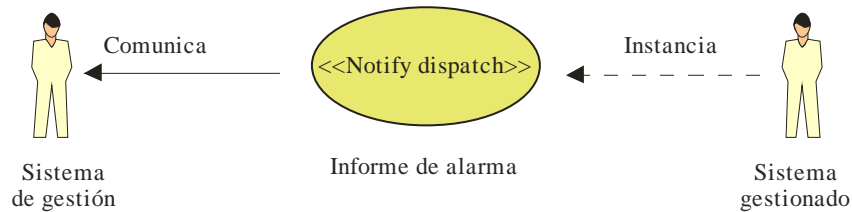
En esta Recomendación, el equipo de red gestionado se considera como los recursos de telecomunicaciones pertinentes.

## 2.4 Diagramas de casos de uso de alto nivel

### 2.4.1 Informe de alarma

El primer diagrama de caso de uso general de la Figura I.1 muestra la interacción global de la interfaz de alarma.

El primer diagrama de caso de uso general muestra las interacciones que se llevan a cabo para informar de un fallo detectado.



M.3020(11)\_E1.1

**Figura I.1 – Informe de alarma**

## 3 Requisitos de nivel de especificación

### 3.1 Requisitos

No hay requisitos de nivel de especificación.

### 3.2 Roles de los actores

Véase la subcláusula 2.2 de esta plantilla.

### 3.3 Recursos de telecomunicaciones

Véase la subcláusula 2.3 de esta plantilla.

### 3.4 Casos de uso

#### 3.4.1 Notificación de fallos

Etapa del caso de uso	Evolución/especificación	Uso relacionado con <<Uses>>
Objetivo <sup>(*)</sup>	Cuando se detecta una condición de fallo, el sistema gestionado envía a través de la interfaz Q una notificación de informe de alarma del tipo pertinente al sistema de gestión.	
Actores y roles <sup>(*)</sup>	El sistema de gestión es consumidor de las notificaciones del sistema gestionado.	
Recursos de telecomunicaciones	Cualquier entidad gestionada	
Supuestos	Se detecta un condición de fallo	
Condiciones <i>a priori</i>	Hay un canal de comunicación abierto entre el sistema de gestión y el sistema gestionado.	
Inicio	Se detecta una condición de fallo	
Etapa 1 <sup>(*)</sup>	Cuando se detecta una condición de fallo, se crea el informe de alarma o informe de alarma de seguridad correspondiente.	



<b>Etapa del caso de uso</b>	<b>Evolución/especificación</b>	<b>Uso relacionado con &lt;&lt;Uses&gt;&gt;</b>
Fin	El agente emite el informe de alarma o el informe de alarma de seguridad.	
Excepciones	Un fallo de la comunicación o del proceso puede dar lugar a un fallo en la entrega del informe de alarma al sistema de gestión. El caso de uso sincronización de alarmas abarca esta posibilidad.	
Condiciones <i>a posteriori</i>	El sistema de gestión queda informado de la condición de fallo en el sistema gestionado.	
Rastreo <sup>(*)</sup>	REQ-FM-FUN-01, REQ-FM-FUN-02, etc.	

### 3.4.2 Anulación de alarmas

...

### 3.4.3 Acuse de recibo de alarmas

...

## Apéndice II

### Ejemplo de análisis

(Este apéndice no forma parte integrante de la presente Recomendación.)

NOTA – El siguiente ejemplo se basa en la gestión de alarmas, pero se utiliza sólo con fines ilustrativos y no pretende ser un conjunto completo o correcto de requisitos para la gestión de alarmas.

#### 1 Conceptos y antecedentes

Toda evaluación de los NE y del estado de salud general de la red necesita que se detecten los fallos en la red y, por consiguiente, se notifiquen las alarmas al OS (EM y/o NM).

...

#### 2 Clases de objeto de información

##### 2.1 Entidades de información importadas y etiquetas locales

Referencia de etiqueta	Etiqueta local
3GPP TS 32.302, clase de objeto de información, NotificationIRP	NotificationIRP
3GPP TS 32.302, interfaz, notificationIRPNotification	NotificationIRPNotification
3GPP TS 32.622, clase de objeto de información, IRPAgent	IRPAgent
3GPP TS 32.312, clase de objeto de información, ManagedGenericIRP	ManagedGenericIRP

##### 2.2 Diagrama de clase

En esta subcláusula se presenta el conjunto de clases de objeto de información (IOC) que encapsulan información en el agente. El objetivo es identificar la información necesaria para que AlarmAgent realice sus operaciones y emita notificaciones. En esta subcláusula se presentan brevemente todas las clases de objeto de soporte en UML. En las siguientes subcláusulas se especifican más detalladamente los diversos aspectos de estas clases de objeto de soporte.

## 2.2.1 Atributos y relaciones

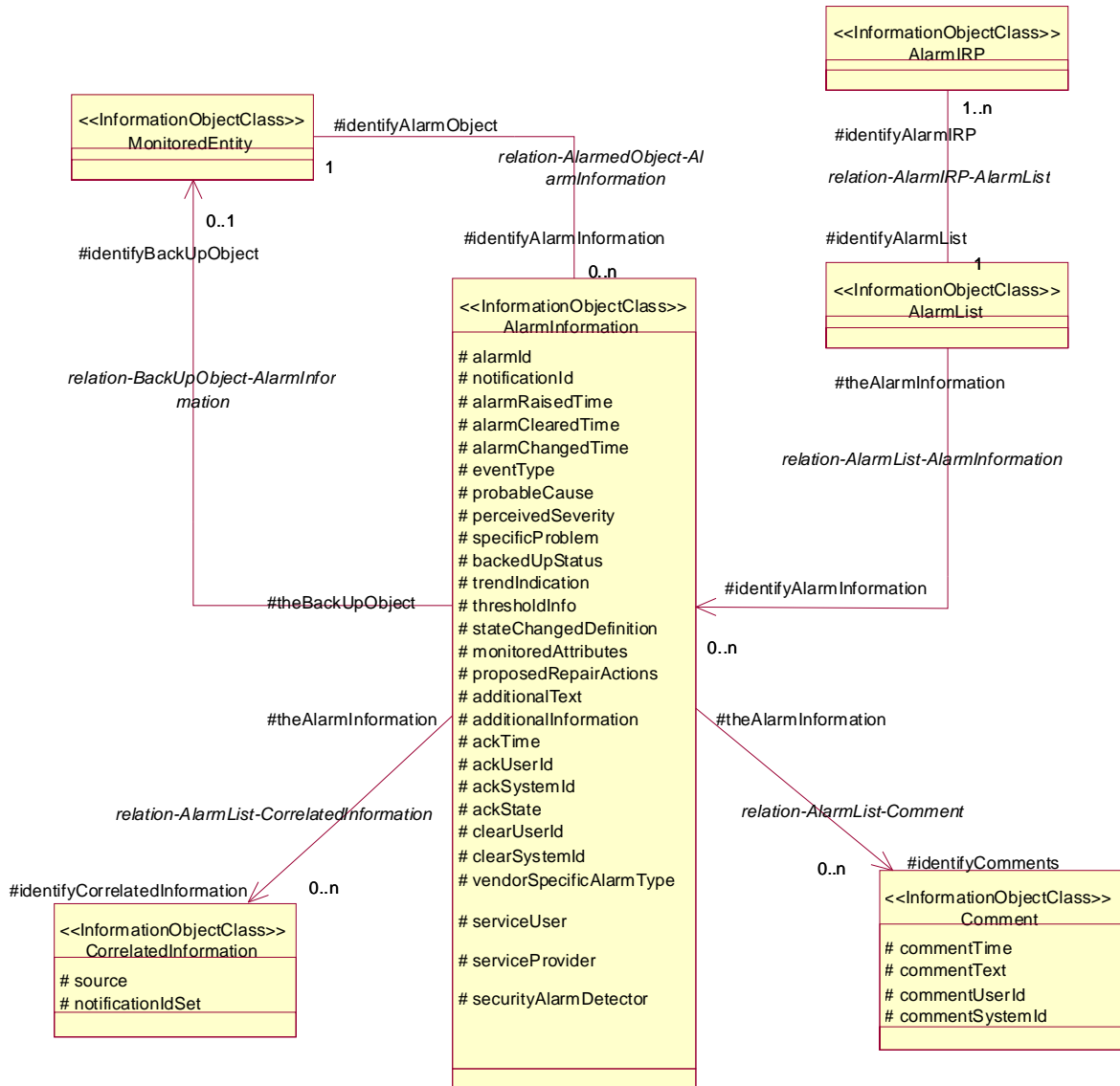


Figura II.1 – Clases de objeto de información de gestión de alarmas

## 2.2.2 Herencia

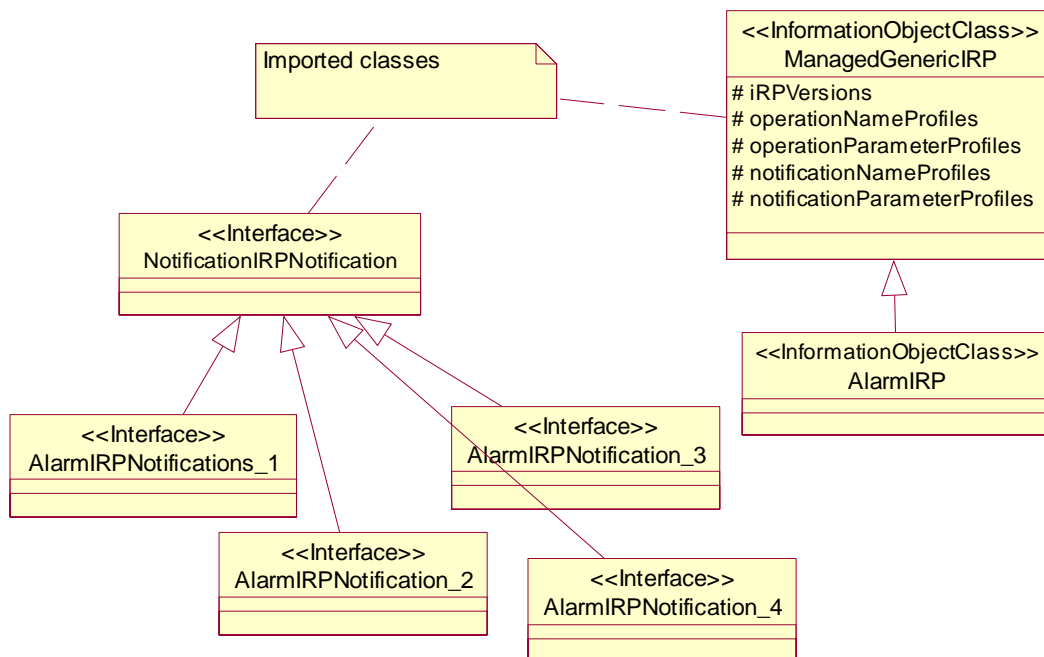


Figura II.2 – Herencia de IOC de gestión de alarmas

## 2.3 Definición de clases de objeto de información

Nombre de clase	Calificativo	ID de requisitos
AlarmInformation	M	REQ-FM-FUN-01, REQ-FM-FUN-02, etc.
AlarmList	M	REQ-FM-FUN-n
...		

### 2.3.1 AlarmInformation

#### 2.3.1.1 Definición

AlarmInformation contiene información sobre una condición de alarma de una MonitoredEntity en situación de alarma.

....

#### 2.3.1.2 Atributos

Nombre de atributo	Calificativo de soporte	Calificativo de lectura	Calificativo de escritura	ID de requisitos
alarmed	M	M	M	
probableCause	C	M	C	
structuredProbableCause	C	M	C	
perceivedSeverity	M	M	M	
specificProblem	O	O	O	
...				
...				

### 2.3.1.3 Diagrama de estados

Las alarmas tienen estados.

...

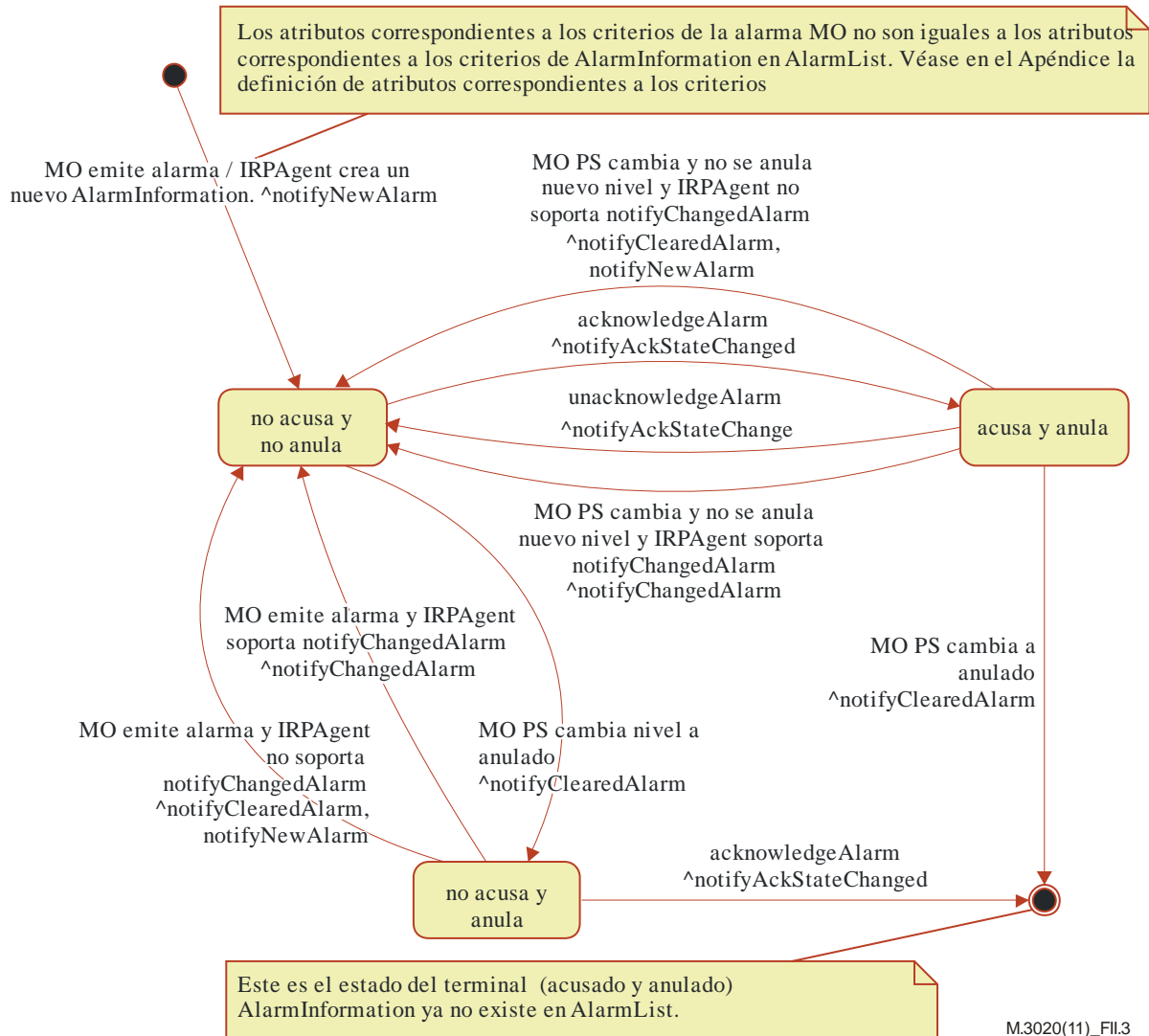


Figura II.3 – Diagrama de estados de información de alarmas

### 2.3.2 AlarmList

### 2.4 Definición de relaciones de información

Relación	Calificativo de soporte	ID de requisitos
relation-AlarmIRP-AlarmList	M	REQ-FM-FUN-x
...		

#### 2.4.1 relation-AlarmIRP-AlarmList (M)

##### 2.4.1.1 Definición

Representa la relación entre AlarmIRP y AlarmList.

### 2.4.1.2 Roles

Nombre	Definición
identifyAlarmIRP	Representa la capacidad de obtener identidades de uno o más AlarmIRP.
identifyAlarmList	Representa la capacidad de obtener la identidad de un AlarmList.

### 2.4.1.3 Restricciones

No hay restricciones para esta relación.

### 2.4.2 relation-AlarmList-AlarmInformation (M)

...

## 2.5 Definición de atributo de información

### 2.5.1 Definición y valores legales

Nombre	Definición	Tipo de información/ valores legales
alarmed	Identifica un AlarmInformation en AlarmList	INTEGER
notificationId	Identifica la notificación que lleva AlarmInformation	INTEGER
ntfSubscription State	Indica el estado de activación de un abono	ENUMERATED/"suspended": se suspende el abono. "notSuspended": el abono está activo.

### 2.5.2 Restricciones

Nombre	Atributos afectados	Definición
inv_notificationId	notificationId	NotificationIds se seleccionará de modo que sea único para todas las notificaciones de un objeto gestionado particular (que representa el NE) durante todo el tiempo que la correlación de alarma sea significativa. El algoritmo mediante el cual se logra la correlación de alarma queda fuera del alcance de este IRP.

### 3 Definición de interfaz

#### 3.1 Interfaces que representan el diagrama de clase

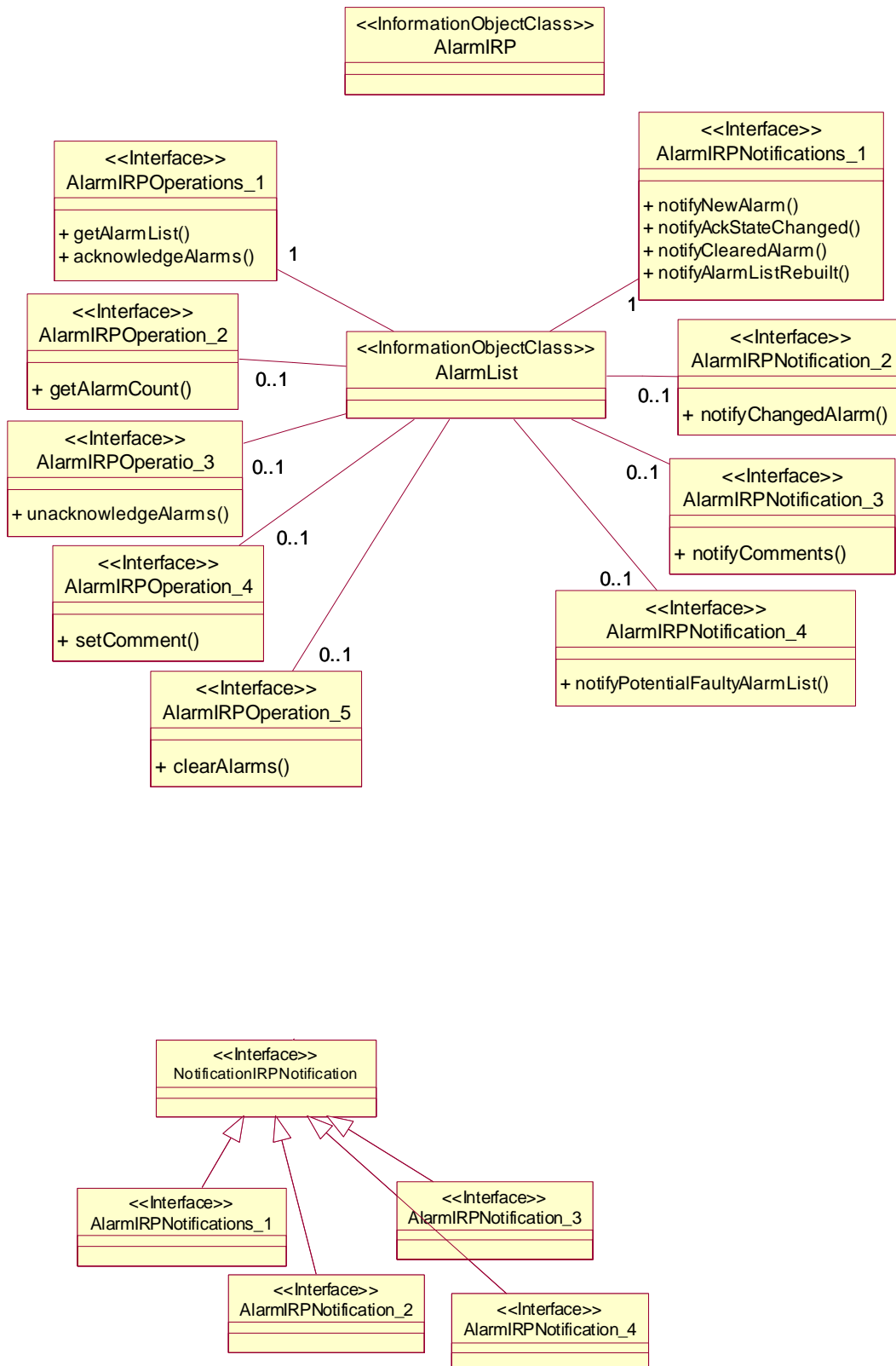


Figura II.4 – Diagrama de clase IRP de gestión de alarmas

### 3.2 Normas genéricas

**Norma 1:** Cada operación con al menos un parámetro de entrada soportará una condición *a priori* `valid_input_parameter`, que indica que todos los parámetros de entrada serán válidos con respecto a su tipo de información. Además, cada una de estas operaciones soportará una excepción `operation_failed_invalid_input_parameter`, creada cuando la condición *a priori* `valid_input_parameter` es falsa. La excepción tiene el mismo estado de entrada y salida.

**Norma 2:** Cada operación con al menos un parámetro de entrada optativo soportará un conjunto de condiciones *a priori* `supported_optional_input_parameter_xxx` donde "xxx" es el nombre del parámetro de entrada optativo y la condición *a priori* indica que la operación soporta el parámetro de entrada optativo nombrado. Además, cada una de estas operaciones soportará una excepción `operation_failed_unsupported_optional_input_parameter_xxx`, creada cuando:

- la condición *a priori* `supported_optional_input_parameter_xxx` es falsa; y
- el parámetro de entrada optativo nombrado transporta información.

La excepción tiene el mismo estado de entrada y salida.

**Norma 3:** Cada operación soportará una excepción genérica `operation_failed_internal_problem`, creada cuando hay un problema interno y no se puede completar la operación. La excepción tiene el mismo estado de entrada y salida.

### 3.3 Interface AlarmIRPOperations\_1 (O)

Nombre de operación	Calificativo	ID de requisitos
<code>acknowledgeAlarms</code>	M	REQ-FM-FUN-x, REQ-FM-FUN-y
<code>getAlarmList</code>	M	...

#### 3.3.1 Operación `acknowledgeAlarms` (M)

##### 3.3.1.1 Definición

El gestor invoca esta operación para acusar recibo de una o más alarmas.

##### 3.3.1.2 Parámetros de entrada

Nombre de parámetro	Calificativo de soporte	Tipo de información/ valores legales	Observaciones
...			
<code>eventIdList</code>	M	SET OF INTEGER/–	Lista de alarmas cuyo recibo se ha de acusar

##### 3.3.1.3 Parámetros de salida

Nombre de parámetro	Calificativo de soporte	Información de correspondencia/ tipo de información/ valores legales	Observaciones
...			
Estado	M	-- / ENUM / "OperationSucceeded": Si <code>allAlarmsAcknowledged</code> es verdadero, "OperationPartiallySucceeded": Si <code>someAlarmAcknowledged</code> es verdadero, "OperationFailed": Si <code>operationFailed</code> es verdadero.	



### 3.3.1.4 Condiciones *a priori*

atLeastOneValidId.

Nombre de afirmación	Definición
atLeastOneValidId	AlarmInformationReferenceList contiene al menos un identificador que identifica un AlarmInformation en AlarmList, y el AlarmInformation identificado tendrá su ackState indicando "unacknowledged" y, en su caso, un perceivedSeverity igual.

### 3.3.1.5 Condiciones *a posteriori*

someAlarmAcknowledged O allAlarmsAcknowledged.

Nombre de afirmación	Definición
someAlarmAcknowledged	...
allAlarmsAcknowledged	...

### 3.3.1.6 Excepciones

Nombre	Definición
operation_failed	<b>Condición:</b> Condición <i>a priori</i> falsa o condición <i>a posteriori</i> falsa. <b>Información devuelta:</b> Estado del parámetro de salida. <b>Estado de salida:</b> Estado de entrada.

### 3.3.2 Operación getAlarmList (M)

...

## Apéndice III

### Comparación con la Recomendación UIT-T Z.601

(Este apéndice no forma parte integrante de la presente Recomendación.)

En este apéndice se presenta información sobre la relación entre esta Recomendación y [b-UIT-T Z.601], utilizada para la elaboración de Recomendaciones de la serie UIT-T M.1400.

Si bien esta Recomendación contiene una metodología para la especificación de interfaces de gestión entre dos sistemas físicos, [b-UIT-T Z.601] ofrece un marco para el desarrollo de un sistema. Esta arquitectura de datos identifica las interfaces candidatas dentro de un sistema, así como las interfaces en las fronteras del mismo. Dichas interfaces están en la frontera entre sistemas.

La metodología especificada en la presente Recomendación tiene por principal objetivo la creación de una serie de Recomendaciones sobre interfaces de gestión, más que de sistemas individuales. La arquitectura de datos no prescribe una captura de requisitos semejante a la fase de requisitos, pues sólo establece la especificación de sistemas individuales y no su objetivo con respecto a una organización.

[b-UIT-T Z.601] se centra en la especificación de la terminología y la gramática externas, tal y como las perciben los usuarios extremos. Esta Recomendación contempla la especificación de interfaces de gestión que los usuarios extremos pueden no percibir.

En esta Recomendación, los requisitos para la solución de problemas se dividen en dos clases. La primera clase de requisitos son los requisitos comerciales y la segunda se denomina requisitos de especificación. Entre los requisitos de especificación se pueden incluir aquellos que soportan la interacción con el usuario extremo en las interfaces persona-máquina. Algunos de estos requisitos pueden especificar requisitos sintácticos que habrán de soportarse en todas las interfaces de gestión. Los requisitos sintácticos se corresponden con los esquemas terminológicos externos de la arquitectura de datos descrita en [b-UIT-T Z.601].

El resultado de la fase de análisis será un modelo de información, lo que se corresponde con un esquema conceptual de la arquitectura de datos de [b-UIT-T Z.601]. Si los modelos de información resultantes de la fase de análisis no transmiten toda la información necesaria a partir de los requisitos sintácticos, es posible que en el diseño de la aplicación se deba incluir una correspondencia con los requisitos sintácticos.

La documentación a partir de la fase de diseño de la aplicación constará de dos partes:

- 1) Una especificación de datos dependiente de la tecnología común a varias interfaces, por ejemplo, utilizando GDMO o CORBA IDL, correspondiente a un esquema de terminología interna conforme con la arquitectura de datos de [b-UIT-T Z.601].
- 2) Una especificación dependiente de la tecnología de cada interfaz, por ejemplo, utilizando CMIP o CORBA IDL, correspondiente a un esquema de distribución conforme con la arquitectura de datos de [b-UIT-T Z.601].

## Apéndice IV

### Temas que quedan en estudio

(Este apéndice no forma parte integrante de la presente Recomendación.)

En este apéndice se indican los temas que quedan en estudio.

#### IV.1 SOA

La aprobación de [UIT-T M.3060] (Principios para la gestión de redes de próxima generación) marcó el cambio de la gestión orientada al objeto a la gestión orientada al servicio. Se ha de estudiar la repercusión que ha tenido este cambio para identificar las modificaciones que se habrán de aportar a esta Recomendación en futuras revisiones.

#### IV.2 UML

En esta versión de UIT-T M.3020 se utiliza UML versión 1.5 a fin de mantener la armonía con las especificaciones 3GPP correspondientes. Las futuras revisiones de UIT-T M.3020 se deberá hacer referencia a las versiones más recientes de UML:

- El meta-metamodelo MOF OMG integra UML 2.x como metamodelo soportado por la mayoría de fabricantes de herramientas. Antes de UML 2.0 no había un meta-metamodelo general y UML ni siquiera estaba normalizado. MOF soporta la adición y creación de nuevos metamodelos definidos de manera precisa a través de OCL, que es un lenguaje de cálculo predicado.
- Tanto la industria (empresas de telecomunicaciones, gobiernos y ejércitos) como los fabricantes de herramientas convergen en el modelo MOF OMG.
- Las ventajas del meta-metamodelo MOF son que soporta una familia de metamodelos que pueden emplearse para definir modelos de objetos, relaciones HCI, diversas aplicaciones específicas de la tecnología y que permite transformaciones entre modelos de manera normalizada. Nada de todo ello se puede conseguir con UML 1.5, pues UML 1.5 es independiente de un metamodelo superior.

#### IV.3 Visibilidad

Se ha sugerido que la visibilidad por defecto sea privada para los atributos y pública para las operaciones a fin de fomentar la encapsulación de los datos y reducir el tiempo y los esfuerzos invertidos en la definición de un modelo de aplicación.

#### IV.4 Definición de tipos

Al elaborar una nueva especificación basada en esta metodología es necesario especificar los tipos de parámetros y atributos. En esta versión de la Recomendación no hay definiciones formales de tipos, por lo que tales definiciones pueden ser diferentes e incoherentes para el mismo significado en distintas especificaciones, por ejemplo, una matriz de enteros puede definirse como una lista de enteros, una secuencia de enteros o un conjunto de enteros.

En el Anexo E se definen los tipos que pueden utilizarse en el modelo conceptual.

## Apéndice V

### Ejemplos de utilización de UML adicionales

(Este apéndice no forma parte integrante de la presente Recomendación.)

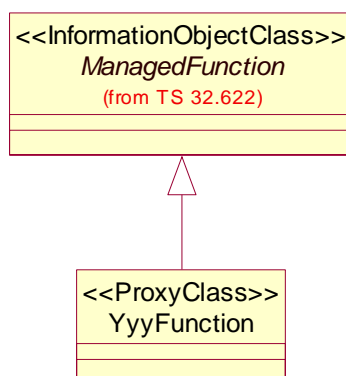
En este apéndice se presentan otros ejemplos de la utilización de UML descrita en el Anexo C.

#### V.1 Clase proxy

##### V.1.1 Primer ejemplo

Se muestra un <<ProxyClass>> denominado YyyFunction. Representa todas las IOC enumeradas en la Nota al pie del diagrama UML. En el contexto de este ejemplo, todas las IOC enumeradas heredan de la IOC ManagedFunction.

La utilización de <<ProxyClass>> elimina la necesidad de dibujar múltiples recuadros <<InformationObjectClass>> UML, es decir, para los nombres enumerados en la Nota, en el diagrama UML.



NOTA – El <<ProxyClass>> YyyFunction representa AsFunction, AucFunction, BgFunction, etc.

#### Ejemplo de notación de <<ProxyClass>> V.1

##### V.1.2 Segundo ejemplo

Se muestra un <<ProxyClass>> denominado YyyFunction. Representa todas las IOC enumeradas en la nota al pie del diagrama UML. En el contexto de este ejemplo, todas las IOC enumeradas tienen relaciones de vínculo (internas y externas).

Los nombres reales de las IOC representadas por el <<ProxyClass>> InternalYyyFunction y por el <<ProxyClass>> ExternalYyyFunction se indican debajo de la subcláusula de X.Y del YyyFunction asociado. Por ejemplo, bajo X.Y.1 para AsFunction, se añaden dos párrafos para enumerar todas las entidades internas y externas pares vinculadas a AsFunction. Véase como ejemplo la siguiente cita donde se utiliza AsFunction como ejemplo para YyyFunction.

Los nombres reales de las IOC representadas por el <<ProxyClass>> Link\_a\_z y el <<ProxyClass>> ExternalLink\_a\_z se indican debajo de la subcláusula de X.Y del YyyFunction asociado. Por ejemplo, bajo X.Y.1 para la AsFunction, se añaden dos párrafos para enumerar los nombres de las IOC representadas por Link\_a\_z y ExternalLink\_a\_z. Véase la siguiente cita donde se utiliza AsFunction como ejemplo para YyyFunction.

"

## X.Y.1 AsFunction

### X.Y.1.1 Definición

Esta IOC representa la funcionalidad As. Puede encontrarse más información sobre As en [b-3GPP TS 23.002].

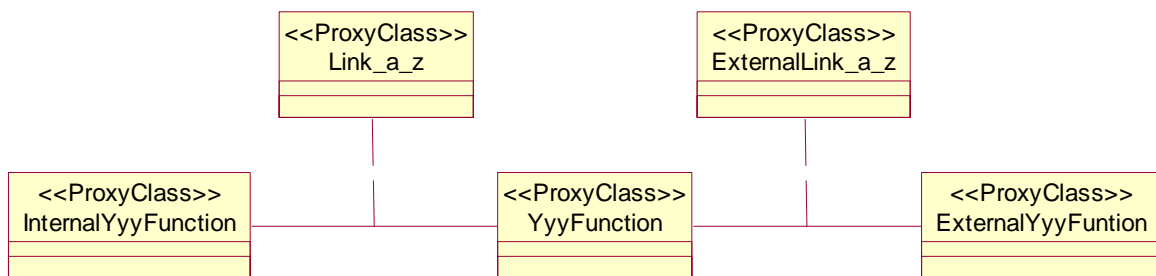
El <<proxyClass>> InternalYyyFunction vinculado representa SlsFunction, CscfFunction, HlrFunction ...

El <<ProxyClass>> ExternalYyyFunction vinculado representa ...

El <<ProxyClass>> Link\_a\_z representa Link\_As\_Scscf, Link\_Bgcf\_Scscf ...

El <<ProxyClass>> ExternalLink\_a\_z representa ...

"



NOTA -'Yyy' en el <<ProxyClass>> YyyFunction representa AsFunction, AucFunction, etc.

### Ejemplo de notación de <<ProxyClass>> V.2

## Apéndice VI

### Directrices sobre numeración de requisitos

(Este apéndice no forma parte integrante de la presente Recomendación.)

El formato de numeración de los requisitos es el siguiente:

REQ-Etiqueta-Categoría-Número

donde "Etiqueta" es la abreviatura de la Recomendación (o parte de la misma). El conjunto de etiquetas es infinito y no se somete a normalización. El conjunto de categorías se define en esta Recomendación.

Problemas:

- ¿Cómo estructurar las etiquetas en especificaciones de requisitos de gran tamaño?
- ¿Qué hacer cuando se suprimen y añaden requisitos?

Las siguientes directrices pueden resultar de utilidad:

- Nunca se han de renumerar los requisitos. La única excepción a esta regla es la primera publicación de una especificación, pero incluso en ese caso conviene evitar la remuneración, pues es posible que la especificación ya se haya utilizado siendo un proyecto.
- Dado que no hay que renumerar los requisitos, no se puede esperar que los requisitos estén numerados secuencialmente en toda la especificación.
- La etiqueta puede utilizarse para dividir la numeración en segmentos lógicos. Por ejemplo, se recomienda utilizar el estilo "A\_B" para identificar "B" como un segmento lógico de "A". Sin embargo, pueden utilizarse otros estilos siempre y cuando se mantenga la estructura de separar con "-" los campos del número de requisito.
- No se recomienda utilizar notaciones con sufijos o prefijos, es decir, añadir algo delante del "Número" o detrás de él, pues no se prevé que la parte "Número" contenga información semántica.
- Como alternativa al estilo "A\_B", los autores de una especificación pueden optar por asignar una gama de números a un grupo de requisitos. Este método debe estar permitido.

## Bibliografía

- [b-UIT-T M.1401] Recomendación UIT-T M.1401 (2006), *Formalización de las designaciones de interconexión entre redes de telecomunicaciones de operadores.*
- [b-UIT-T M.1403] Recomendación UIT-T M.1403 (2007), *Formalización de encargos genéricos.*
- [b-UIT-T M.1404] Recomendación UIT-T M.1404 (2007), *Formalización de encargos para interconexiones de redes de operadores.*
- [b-UIT-T Z.601] Recomendación UIT-T Z.601 (2007), *Arquitectura de datos de un sistema de software.*
- [b-3GPP TS 23.002] 3GPP TS 23.002 (in force), *Network architecture.*
- [b-3GPP TS 32.101] 3GPP TS 32.101 V10.0.0 (2010), *Telecommunication management; Principles and high level requirements.*
- [b-3GPP TS 32.150] 3GPP TS 32.150 V10.2.0 (2011), *Telecommunication management; Integration Reference Point (IRP) Concept and definitions.*
- [b-3GPP TS 32.151] 3GPP TS 32.151 V10.1.0 (2010), *Telecommunication management; Integration Reference Point (IRP) Information Service (IS) template.*
- [b-3GPP TS 32.152] 3GPP TS 32.152 V10.0.0 (2010), *Telecommunication management; Integration Reference Point (IRP) Information Service (IS) Unified Modelling Language (UML) repertoire.*
- [b-3GPP TS 32.302] 3GPP TS 32.302 V10.0.0 (2010), *Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP); Information Service (IS).*





## SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedia
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedia
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
<b>Serie M</b>	<b>Gestión de las telecomunicaciones, incluida la RGT y el mantenimiento de redes</b>
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Terminales y métodos de evaluación subjetivos y objetivos
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, comunicaciones de sistemas abiertos y seguridad
Serie Y	Infraestructura mundial de la información, aspectos del protocolo Internet y redes de la próxima generación
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación