



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

Q.65

(06/2000)

SERIE Q: CONMUTACIÓN Y SEÑALIZACIÓN

Funciones y flujos de información para servicios de la
RDSI – Metodología

**Metodología funcional unificada para la
caracterización de servicios y capacidades
de red, incluyendo técnicas alternativas
orientadas a objetos**

Recomendación UIT-T Q.65

(Anteriormente Recomendación del CCITT)

RECOMENDACIONES UIT-T DE LA SERIE Q
CONMUTACIÓN Y SEÑALIZACIÓN

SEÑALIZACIÓN EN EL SERVICIO MANUAL INTERNACIONAL	Q.1–Q.3
EXPLOTACIÓN INTERNACIONAL SEMIAUTOMÁTICA Y AUTOMÁTICA	Q.4–Q.59
FUNCIONES Y FLUJOS DE INFORMACIÓN PARA SERVICIOS DE LA RDSI	Q.60–Q.99
Metodología	Q.60–Q.67
Servicios básicos	Q.68–Q.79
Servicios suplementarios	Q.80–Q.99
CLÁUSULAS APLICABLES A TODOS LOS SISTEMAS NORMALIZADOS DEL UIT-T	Q.100–Q.119
ESPECIFICACIONES DE LOS SISTEMAS DE SEÑALIZACIÓN N.º 4 Y N.º 5	Q.120–Q.249
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.º 6	Q.250–Q.309
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN R1	Q.310–Q.399
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN R2	Q.400–Q.499
CENTRALES DIGITALES	Q.500–Q.599
INTERFUNCIONAMIENTO DE LOS SISTEMAS DE SEÑALIZACIÓN	Q.600–Q.699
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.º 7	Q.700–Q.849
SISTEMA DE SEÑALIZACIÓN DIGITAL DE ABONADO N.º 1	Q.850–Q.999
RED MÓVIL TERRESTRE PÚBLICA	Q.1000–Q.1099
INTERFUNCIONAMIENTO CON SISTEMAS MÓVILES POR SATÉLITE	Q.1100–Q.1199
RED INTELIGENTE	Q.1200–Q.1699
REQUISITOS Y PROTOCOLOS DE SEÑALIZACIÓN PARA IMT-2000	Q.1700–Q.1799
RED DIGITAL DE SERVICIOS INTEGRADOS DE BANDA ANCHA (RDSI-BA)	Q.2000–Q.2999

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

Recomendación UIT-T Q.65

Metodología funcional unificada para la caracterización de servicios y capacidades de red, incluyendo técnicas alternativas orientadas a objetos

Resumen

Esta Recomendación UIT-T contiene la metodología funcional unificada (UFM) que sustituye a UIT-T Q.65 (junio de 1997) existente hasta ahora; describe una arquitectura funcional común para la prestación de servicios y trata de los requisitos de señalización de la implantación de servicios. El método general de elaboración de Recomendaciones de conmutación y de señalización para servicios de la RDSI, que consta de tres etapas, se describe en UIT-T I.130. El método se ha generalizado más allá de la RDSI para que incluya servicios prestados por redes de diversos tipos, y entre las mismas. La metodología funcional unificada combina el planteamiento tradicional de la versión de 1988 de esta Recomendación con algunos de los planteamientos tradicionalmente utilizados en el método de descripción de la red inteligente. El método detallado para obtener la parte de la etapa 2 de estas Recomendaciones se describe en esta Recomendación. El texto principal de esta Recomendación presenta información básica sobre la metodología funcional unificada, los pasos de este método, los convenios utilizados en técnicas de descripción, y las directrices de utilización.

Orígenes

La Recomendación UIT-T Q.65, revisada por la Comisión de Estudio 11 (1997-2000) del UIT-T, fue aprobada por el procedimiento de la Resolución 1 de la CMNT el 15 de junio de 2000.

PREFACIO

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

La Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), 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 CMNT.

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

NOTA

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

PROPIEDAD INTELECTUAL

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

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

© UIT 2001

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

ÍNDICE

Página

1	Introducción	1
1.1	Metodología funcional unificada (UFM): Sumario	1
1.2	Definición de la etapa 2 (sumario).....	5
2	Pasos del método	6
2.1	Paso 1 – Modelo funcional	6
2.1.1	Modelo funcional unificado.....	7
2.1.2	Entidades funcionales	7
2.1.3	Relaciones entre entidades funcionales	15
2.1.4	Diseño del modelo funcional.....	15
2.1.5	Relación entre los modelos de los servicios básico y suplementario	16
2.2	Paso 2 (opcional) – Descripción SIB de las características de los servicios (véase en 3.1, el paso 2 alternativo que utiliza técnicas orientadas a objetos).....	17
2.2.1	Definiciones de SIB.....	19
2.2.2	Parámetros de datos SIB	20
2.2.3	Convenios de modelado de SIB.....	21
2.2.4	Modelado SIB de características de servicio	22
2.2.5	Lista de SIB disponibles	23
2.2.6	Correspondencia de SIB con FE	23
2.3	Paso 3 – Diagramas de flujo de información (véase en 3.2 el paso 3 alternativo que utiliza técnicas orientadas a objetos).....	24
2.3.1	Identificación de los flujos de información	24
2.3.2	Definición de los distintos flujos de información.....	26
2.4	Paso 4 – Acciones de las entidades funcionales (véase en 3.3 el paso 4 alternativo que utiliza técnicas orientadas a objetos).....	27
2.5	Paso 5 (opcional) – Diagramas SDL para entidades funcionales	27
2.5.1	Aspectos generales del SDL	28
2.5.2	Adición de funcionalidad a las descripciones de servicios de la etapa 2 existentes	38
2.5.3	Lotes	43
2.6	Paso 6 – Asignación de las entidades funcionales a emplazamientos físicos (escenarios)	43
3	Pasos alternativos que utilizan técnicas orientadas a objetos	44
3.1	Paso alternativo 2.....	45
3.2	Paso alternativo 3.....	46
3.2.1	Clases interfaz.....	46
3.2.2	Diagramas de clase	47
3.2.3	Diagramas de secuencias de interfaz (opcional).....	49
3.3	Paso alternativo 4.....	50

Apéndice I – Formato y estructura de una descripción de la etapa 2 utilizando la metodología funcional unificada.....	51
Apéndice II – Arquitectura funcional – Evolución Q.65	52
Apéndice III – Ejemplos de clases interfaz formadas a partir de requerimientos de servicio/aplicación	54
III.1 Introducción	54
III.2 Ejemplo de servicio/aplicación.....	54
III.2.1 Llamada iniciada por terceros.....	54
III.3 Diagramas de clase	55
III.3.1 Diagrama de clase de servicio genérico de control de llamada	55
III.3.2 Invocaciones de método	56
III.3.3 IDL.....	56
III.4 ¿A dónde ir ahora?	58

Recomendación UIT-T Q.65

Metodología funcional unificada para la caracterización de servicios y capacidades de red, incluyendo técnicas alternativas orientadas a objetos

1 Introducción

Esta nueva Recomendación contiene un resumen de la metodología funcional unificada (UFM, *unified functional methodology*), que describe una arquitectura funcional común para la prestación de servicios y trata los requisitos de señalización de la implementación de servicios. El método general de elaboración de Recomendaciones de conmutación y de señalización para servicios RDSI, compuesto de tres etapas, se describe en UIT-T I.130. El método se ha generalizado más allá de la RDSI para que incluya servicios prestados en otras redes de diversos tipos, y entre las mismas. Esta nueva UFM combina el planteamiento tradicional de la versión de 1988 de esta Recomendación con algunos de los planteamientos tradicionalmente utilizados en el método de descripción de la red inteligente (RI) y los utilizados en la metodología funcional unificada. Le da al usuario de la Recomendación la posibilidad de utilizar los métodos descritos en la versión de 1997 de UIT-T Q.65 o de adoptar técnicas utilizadas en el sector informático. El método detallado para obtener la parte de la etapa 2 de estas Recomendaciones se describe en esta Recomendación.

El texto principal de esta Recomendación presenta información básica sobre la metodología funcional unificada, los pasos del método, los convenios utilizados en las técnicas de descripción y las directrices de utilización. El apéndice I presenta una estructura propuesta para la redacción de una descripción de servicios de la etapa 2 utilizando la metodología SIB. El apéndice II contiene el modelo funcional unificado incluyendo llamada de banda ancha y separación de portadoras. El apéndice III contiene "Ejemplos de clases de interfaz formadas a partir de requisitos de aplicación/servicio".

1.1 Metodología funcional unificada (UFM): Sumario

La metodología funcional unificada (UFM) permite hacer descripciones funcionales de los servicios sea con flujos de información, acciones de entidades funcionales (FEA, *functional entity actions*) y lenguaje de especificación y descripción (SDL, *specification description language*) o bien descripciones de clases de interfaz, descripciones genéricas de IDL (IDL, *interface description language*) y lenguaje de especificación y descripción (SDL) a partir de una única arquitectura funcional unificada. En el primer caso el concepto de bloques de construcción independientes del servicio (SIB, *service independent building blocks*) ha sido adoptado para tratar las necesidades de creación de servicios, así como para introducir bloques reutilizables de flujos, SDL y FEA que puedan ser catalogados. El usuario tiene también la libertad de usar el lenguaje unificado de modelado (UML, *unified modeling language*) junto con la descripción completa de las descripciones de clase de interfaz para generar invocaciones de método que se corresponden a las API y las IDL. Este nuevo concepto facilita más la generación de SDL. El usuario puede utilizar ambos métodos, uno junto al otro, utilizando el método que mejor facilita la descripción de una interfaz en particular. El modelo unificado permite a todas las arquitecturas de red (por ejemplo, RDSI, RDSI-BA, RI, IMT-2000 y RGT) ser descritas de manera similar.

La metodología necesita tratar las actuales necesidades, pero también debe evolucionar para que incluya las mejoras y la nueva tecnología. El método incluye muchos elementos de la versión de 1988 de esta Recomendación, por lo que puede utilizarse inmediatamente, incluso a medida que evoluciona. El uso del concepto de SIB o de las técnicas orientadas a objeto (O-O) para describir la relación entre objetos racionaliza el trabajo de definición de servicios de la etapa 2. Se crearán

nuevos SIB clases de interfaz a medida que sean necesarios. Se mantendrá un catálogo de SIB clases de interfaz como referencia para la creación y definición de servicios.

Los principios que definen el alcance de la UFM incluyen:

- 1) La metodología funcional unificada permite la creación de descripciones funcionales en las que hay disponibles descripciones de servicios de la etapa 1 o en las que se especifican capacidades de red para la creación de servicios.
- 2) La metodología se basa principalmente en la experiencia adquirida empleando la metodología de la etapa 2 de la RDSI [UIT-T Q.65 (1988)], metodología de la RI [UIT-T Q.1210 (1995)], y técnicas de computación distribuida derivadas de Rumbaugh, Booch y otros, y refinadas por medio del uso de la UML.
 - La Recomendación UIT-T Q.65 (1997) proporciona buena parte de los fundamentos de la descripción de servicios utilizando un modelo funcional, entidades funcionales, y acciones de entidades funcionales.
 - El método de la RI proporciona la flexibilidad, la independencia de los servicios y las características de reutilización que son de desear.
 - El UML es la herramienta a través de la cual se puede desarrollar una descripción completa de las clases de interfaz junto con las relaciones que los ligan. Esto, junto con varios tipos de herramientas que son estándar, facilita generar el IDL que conforma la interfaz de programación de aplicación (API, *application programming interface*).
- 3) El método se basa en un modelo funcional unificado.
 - La arquitectura funcional se expande para incluir nuevos requisitos y puede expandirse aún más para adaptarse a las necesidades de los usuarios.
 - En cada descripción de servicio, se selecciona un conjunto apropiado de entidades funcionales para componer el modelo.
 - Esto conduce a un conjunto coherente de flujos y/o API y diagramas SDL.
- 4) Mantener una biblioteca de SIB (con los correspondientes flujos de información y SDL) que se utilizan para determinar la arquitectura funcional en el plano funcional distribuido (véase la figura 1).
 - Se están creando conjuntos de SIB para soportar los conjuntos de capacidades de las Recomendaciones de la RI.
 - Los SIB se ponen en correspondencia con la estructura funcional completa con acciones de entidades funcionales predefinidas, así como un SDL y flujos de información predefinidos.
 - Los SIB pueden ser herramientas de utilidad para la creación de la etapa 2; otros aspectos (por ejemplo, descripciones SDL, flujos de información) pueden automatizarse potencialmente.
- 5) Mantener una biblioteca de paquetes, clases de interfaz e IDL, que define a la API a través de una determinada interfaz. Éstos se usan también para determinar la arquitectura funcional en el plano funcional distribuido (véase la figura 1).
- 6) La metodología trata las necesidades actuales, pero debe evolucionar para incluir mejoras y nueva tecnología.
 - El método puede utilizarse incluso a medida que evoluciona.
 - Planteamiento escalonado para su utilización en el UIT-T, utilizando la arquitectura unificada.
 - Se necesitarán nuevos SIB (servicios existentes, nuevos servicios proyectados).
 - Se desarrollarán nuevas llamadas de API.

- Es necesario definir nuevas relaciones.
 - Deben considerarse nuevos servicios proyectados (por ejemplo, llamadas multipartitas).
 - Sin ser aún un proceso automatizado, la reutilización desempeña un gran papel.
 - Deben desarrollarse nuevas vías experimentales para describir servicios a partir de SIB.
 - La verificación adicional de los servicios existentes será de utilidad (para comprobar la exactitud del método y la curva de aprendizaje).
- 7) La metodología unificada para crear una descripción de la etapa 2 debe incluir lo siguiente:
- Identificación del servicio o de la capacidad de red.
 - Modelo funcional y definición de entidad funcional (modelo funcional unificado).
 - Identificación de los SIB y/o de paquete (opcional).
 - Acciones de entidad funcional y/o IDL–definición.
 - Flujos de información y/o API.
 - SDL (descripción dinámica) (opcional).
 - Escenarios (atribución física de entidades funcionales).
- 8) Definir escenarios físicos en un momento apropiado.
- Deben adoptarse decisiones en cuanto a las implementaciones de red recomendadas.
 - Con la extensión a la RI, RDSI-BA, IMT-2000, y a otras redes, se aumenta el catálogo de entidades físicas.
 - Esto aumenta en gran medida el número de escenarios potenciales.
- 9) Trabajos para proporcionar Recomendaciones relativas a mejoras de las herramientas y automatización del proceso de descripción funcional.
- Las Recomendaciones UIT-T Z.100 (SDL) y UIT-T Z.120 (Gráficos de secuencias de mensajes – Flujos de información) pueden utilizarse con más eficacia.
 - Las herramientas pueden proporcionar validación del SDL y de los flujos.
 - Comunicación de reglas y convenios para la definición y combinación de SIB; puede ser posible cierta automatización de este proceso.

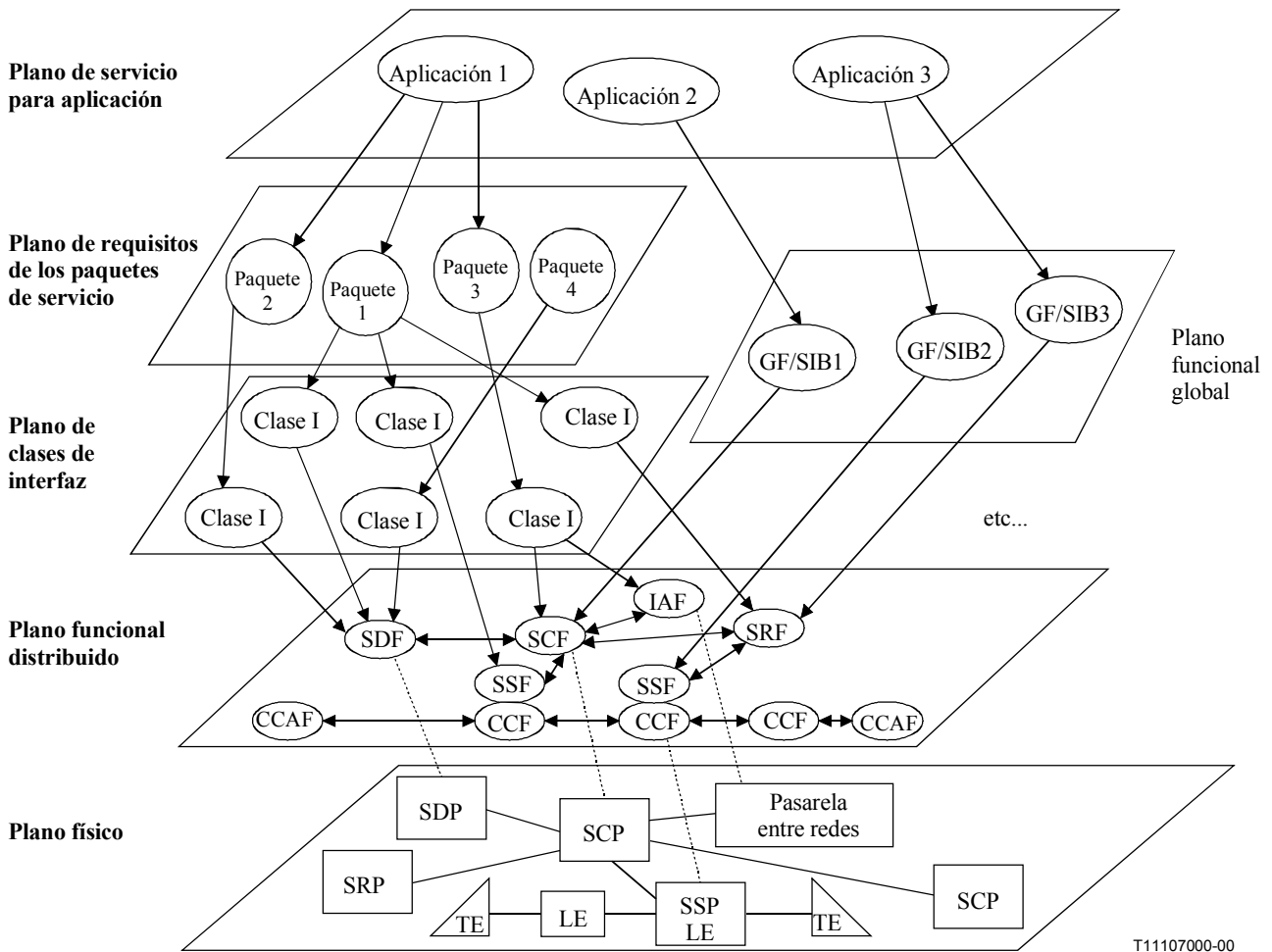


Figura 1/Q.65 – Modelo de metodología conceptual que muestra las relaciones entre los servicios, funciones globales/bloques de construcción independientes del servicio, paquetes de servicios, clases de interfaz, entidades funcionales y entidades físicas

La figura 1 muestra el modelo funcional de RI modificado. El objetivo del modelo funcional es darle al usuario una opción entre aplicar el planteamiento de SIB o utilizar métodos orientados a objetos. Los resultados son relativamente parecidos. Si se utiliza la trayectoria SIB su resultado es flujos de información que tienen su correspondiente en entidades funcionales, mientras que si se utiliza el método orientado a objetos se obtiene como resultado un conjunto de llamadas de API que también tiene su correspondencia en entidades funcionales. Esto se explicará más tarde en la cláusula 3.

Todos los FE descritos dentro del DFP de la figura 1 pueden residir en un único dominio de red a excepción de la función de acceso inteligente (IAF, *intelligence access function*), que puede ser utilizada por otra red no-RI para reflejar la funcionalidad provista por la SCF en la RI. Es por ello que se necesita proveer algún tipo de funcionalidad de pasarela para proteger la división entre las dos redes. La funcionalidad de pasarela puede proveer seguridad y protección cortafuegos, así como capacidades de correspondencia entre protocolos. Por esta razón la figura 1 muestra el equivalente en el plano físico como una pasarela de interconexión entre redes. Este punto de ingreso o egreso de la RI provee también el acceso entre las redes RI e IP.

1.2 Definición de la etapa 2 (sumario)

- especifica un modelo funcional utilizando entidades funcionales (FE, *functional entities*) del modelo funcional unificado para la descripción de un determinado servicio o capacidad de red;
- especifica las acciones de entidad funcional (FEA) necesarias;
- especifica flujos de información o llamadas a API entre FE;
- especifica la descripción SDL de cada FE;
- recomienda un pequeño conjunto de escenarios realistas para la asignación de FE a entidades físicas.

Para los servicios y capacidades de red a normalizar, la etapa 2 del método se sirve de las descripciones de la etapa 1 de servicios básicos y suplementarios y capacidades de red. Para servicios y capacidades de red sin una descripción detallada de la etapa 1, el método de la etapa 2 puede utilizar como entrada descripciones mediante SIB o requisitos de paquete de servicios de características de servicios. La descripción de la etapa 1 considera la red (este término, dentro de este contexto, podría incluir alguna capacidad del equipo del usuario) como una entidad individual que suministra estos servicios al usuario. La descripción de la etapa 2 define las funciones necesarias, así como su distribución dentro de la red. Las interacciones usuario/red de la etapa 1 se utilizan e interpretan en la etapa 2 como se ilustra en la figura 2.

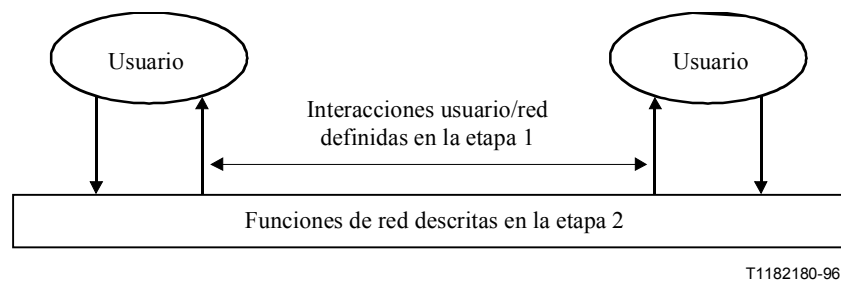


Figura 2/Q.65 – Relación etapa 1/etapa 2

La etapa 2 del método utiliza técnicas que proporcionan las características deseables siguientes:

- especificación funcional única que puede aplicarse a diversas realizaciones físicas diferentes para proporcionar el servicio;
- definición precisa de las capacidades funcionales y su posible distribución entre los equipos de la red (y, en algunos casos, en el equipo del usuario), para soportar servicios básicos y suplementarios y capacidades de red;
- descripción detallada de las funciones, flujos de información y llamadas a API que deben proporcionarse, pero no de la manera de llevarlos a efecto;
- requisitos de los protocolos y capacidades de conmutación así como datos de entrada para la etapa 3 del método.

La salida de la etapa 2 es utilizada por:

- a) diseñadores de protocolos en la etapa 3, para especificar los protocolos que se han de utilizar en entidades físicas separadas;
- b) diseñadores de conmutadores (y otros nodos) para especificar los requisitos funcionales de estos nodos; y
- c) planificadores de redes.

Esta Recomendación describe detalladamente los seis pasos de la etapa 2. El orden de estos pasos representa una aplicación idealizada del método; sin embargo, en la práctica será necesario definir completamente por medio de iteraciones los resultados de la etapa 2. El apéndice I contiene una estructura propuesta de la etapa 2 utilizando la metodología funcional unificada.

Los siguientes pasos difieren ligeramente en función de la utilización del método SIB o del método orientado a objetos, O-O (UML). La mayor parte de los aspectos del paso 1 son iguales independientemente del método utilizado. En la cláusula 3 se detallan los métodos UML y se proporcionan ejemplos cuando corresponda. En el apéndice III se proporciona un ejemplo práctico del uso de UML.

2 Pasos del método

2.1 Paso 1 – Modelo funcional

Para cada servicio básico y suplementario se establece un modelo funcional o una capacidad de red. En cada caso, el modelo se ajusta a los requisitos y características del servicio de que se trata. El modelo funcional utilizado en la descripción de la etapa 2 de un servicio identifica las entidades funcionales así como las relaciones entre ellas. El perfeccionamiento del modelo funcional inicial es realizado a través de la ampliación y/o interacción de los pasos 2 a 6, tal como se describe más adelante. El modelo funcional final representa el resultado de la terminación de la etapa 2.

Como estas Recomendaciones proporcionan al usuario la posibilidad de utilizar sea el método SIB como el método orientado a objetos, se dan a continuación alternativas para los pasos 2-4. En la figura 2.1 se muestran los pasos a seguir cuando se utiliza el método SIB y cuando se usa el método orientado a objetos.

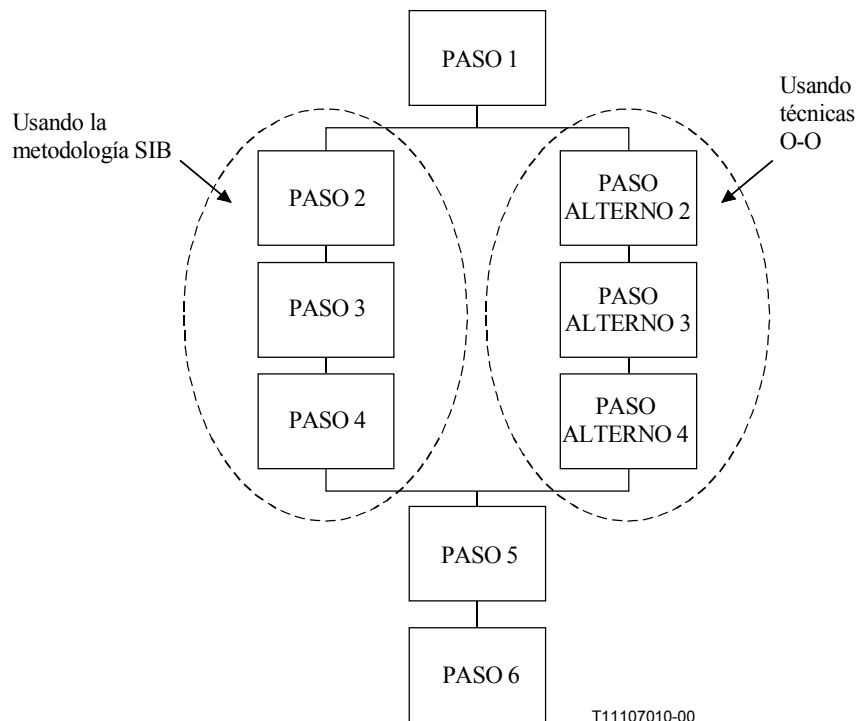


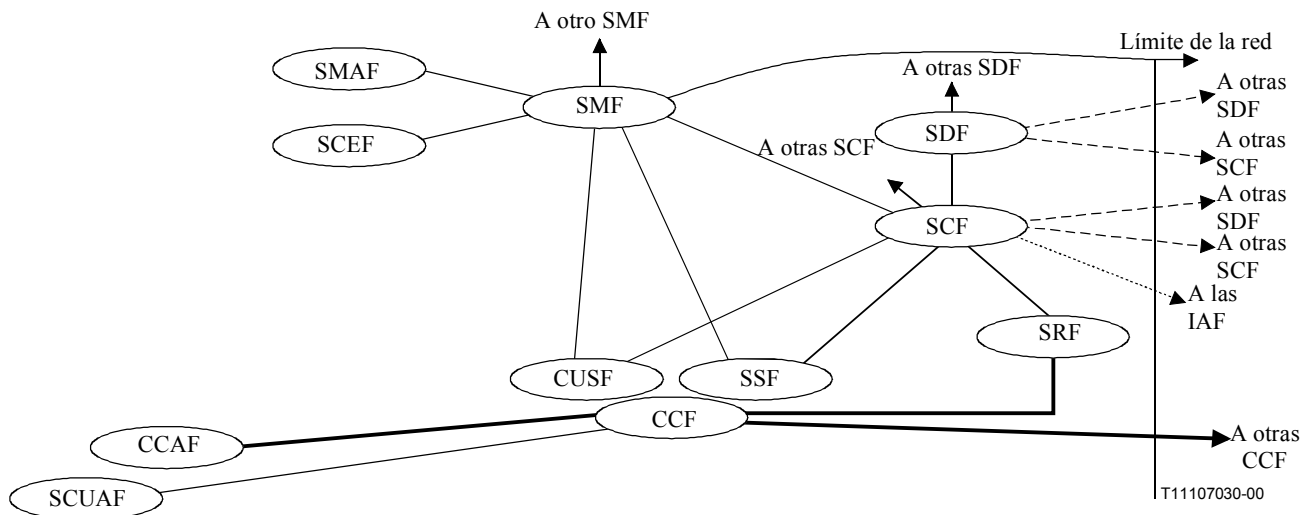
Figura 2-1 – Uso de los pasos 1 a 6

2.1.1 Modelo funcional unificado

El modelo funcional unificado para la metodología de la etapa 2 tiene en cuenta las entidades funcionales comunes, relaciones y flujos de información que surgen de la RDSI, RI, RDSI-BA, IMT-2000 y RGT.

La figura 3 muestra una instancia del modelo funcional unificado derivado del conjunto de FE especificadas por la metodología funcional unificada. El modelo funcional identifica y denomina las distintas FE y sus tipos. También identifica la relación y los tipos de relación entre FE comunicantes. Las entidades funcionales se representan con círculos y la relación entre dos FE comunicantes se identifica por una línea que las une.

NOTA – El modelo de la figura 3 es un modelo compuesto basado en la llamada básica (Q.71), CS-2 de RI, IMT-2000 y RDSI-BA. Este modelo se presenta como base a partir de la cual puede construirse un modelo funcional concreto de la etapa 2 para soportar un determinado servicio o capacidad de red. El apéndice II representa una visión actual de la integración de la RI y la RDSI-BA con plena separación del control de llamada y de conexión. Este tipo de modelo constituirá la base de la evolución de esta Recomendación.



CCAF	Función de agente de control de llamada (<i>call control agent function</i>)
SCF	Función de control del servicio (<i>service control function</i>)
SRF	Función de recursos especializados (<i>specialized resource function</i>)
CCF	Función de control de la llamada (<i>call control function</i>)

IAF	Función de acceso inteligente (<i>intelligent access function</i>)
SCUAF	Función de acceso de usuario al control de servicio (<i>service control user access function</i>)
SSF	Función de conmutación del servicio (<i>service switching function</i>)
SDF	Función de datos del servicio (<i>service data function</i>)

Figura 3/Q.65 – Modelo funcional unificado

2.1.2 Entidades funcionales

Las entidades funcionales reflejan inicialmente una comprensión general de las funciones de red necesarias para prestar el servicio. Las entidades funcionales se definen de la siguiente manera:

- Una entidad funcional es una agrupación de funciones que suministran servicio en una localización particular, y es un subconjunto del conjunto completo de las funciones necesarias para suministrar el servicio.
- Se describe una entidad funcional en términos del control de una instancia de un servicio (por ejemplo, una llamada o una conexión).

- Una entidad funcional es visible para las otras entidades funcionales que necesitan comunicar con ella para proporcionar un servicio (es decir, las entidades funcionales son entidades que pueden ser direccionadas por la red).
- Un modelo funcional puede contener entidades funcionales de diferentes tipos. El tipo de una entidad funcional se caracteriza por el agrupamiento especial de las funciones que lo componen. Así, se dice que dos o más entidades funcionales son del mismo tipo si están formadas por el mismo agrupamiento de funciones.
- Se define normalmente un tipo de entidad funcional independiente para cada agrupación de funciones que pueden estar distribuidas entre equipos físicos diferentes. Sin embargo, si hay un alto grado de aspectos comunes entre las agrupaciones necesarias, puede ser conveniente su definición como subconjuntos de un tipo único en lugar de como tipos diferentes.
- Se determinan entidades funcionales para cada servicio básico y suplementario o capacidad de red. El mismo tipo de entidad funcional puede presentarse más de una vez en un modelo funcional, y también puede aparecer en el modelo de más de un servicio.

Las siguientes son definiciones de las FE utilizadas en el modelo funcional unificado. Muchas de estas FE se han derivado de los estudios RI en los servicios soporte de la red.

2.1.2.1 Función de agente de control de llamada (CCAF, *call control agent function*); función CCA

Función de agente de control de llamada (CCA, *call control agent*) que proporciona acceso a los usuarios. Es la interfaz entre el usuario y las funciones de control de llamada de la red. Esta función:

- a) proporciona acceso de usuario, interactuando con el usuario para establecer, mantener, modificar y liberar, si es necesario, una llamada o instancia de servicio;
- b) da acceso a las capacidades de prestación del servicio de la función de control de llamada (CCF), utilizando peticiones de servicio (por ejemplo, establecimiento, transferencia, retención, etc.) para el establecimiento, manipulación y liberación de una llamada o instancia de servicio;
- c) recibe de la CCF indicaciones relativas a la llamada o servicio y las retransmite al usuario si es necesario;
- d) mantiene la información de estado de la llamada o servicio percibida por la entidad funcional;
- e) ejerce de interfaz a la SCUAF para servicios no relativos a la llamada, de ser necesario.

2.1.2.2 Función de control de llamada (CCF, *call control function*); función CC

Función de control de la llamada (CC, *call control*) de la red que proporciona procesamiento y control de la llamada/servicio. Esta función:

- a) establece, manipula y libera la llamada/conexión como lo "solicita" la SCUAF;
- b) proporciona la capacidad de asociar y relacionar entidades funcionales de SCUAF que intervienen en una determinada llamada y/o instancia de conexión (que puede deberse a peticiones de CUSF);
- c) gestiona la relación entre entidades funcionales de SCUAF que intervienen en una asociación (por ejemplo, supervisa la perspectiva global de la asociación y/o instancia de conexión);
- d) proporciona mecanismos de accionamiento para acceder a la funcionalidad de RI (por ejemplo, cursa eventos a la SSF/CUSF);
- e) gestiona datos básicos de recursos de llamada (por ejemplo, referencias de llamadas).

2.1.2.3 Función de conmutación de servicio (SSF, *service switching function*); función SS

Función de conmutación de servicio, que asociada con la CCF proporciona un conjunto de funciones necesarias para la interacción entre la CCF y una función de control de servicio (SCF), y que está asociada, de ser necesario, a la NCSF para la gestión del servicio no relativo a la llamada. Esta función:

- a) extiende la lógica de la CCF para incluir el reconocimiento de los accionadores del control del servicio y para interactuar con la SCF;
- b) gestiona la señalización entre la CCF y la SCF;
- c) modifica las funciones de procesamiento de la llamada/conexión (en la CCF) si es necesario para procesar peticiones de utilización de servicio proporcionado por la RI bajo el control de la SCF;
- d) establece una interfaz con la CUSF para tratar interacciones no relacionadas con la llamada;
- e) soporta el uso de retransmisión, en cuyo caso asegura que se retransmita la información entre la SCF y la SFR; es posible que lo haga utilizando las capacidades de interacción de usuario relacionada con la llamada por el canal de salida (OCCRUI, *out-channel call related user interaction*).

2.1.2.4 Función de control de servicio (SCF, *service control function*); función SC

Función que ordena funciones de control de la llamada en el procesamiento de peticiones de servicio proporcionado por la red inteligente y/o personalizado. La SCF puede interactuar con otras entidades funcionales para acceder a lógica adicional u obtener información (datos de servicio o de usuario) requeridos para procesar una instancia de lógica de llamada/servicio. Esta función:

- a) ejerce de interfaz e interactúa con las entidades funcionales: función de conmutación del servicio/función de control de la llamada, función de recursos especializados (SRF), función de datos del servicio (SDF), otras funciones de control de servicio (SCF), funciones de acceso inteligente (IAF) y funciones de servicio no relacionados con la llamada (CUSF);
- b) contiene la lógica y la capacidad de procesamiento requeridas para tratar intentos de servicio proporcionados por la RI, tanto relacionados con la llamada como los que no lo son;
- c) ejerce de interfaz e interactúa con otras SCF en modo seguro para control de servicios distribuido y para las notificaciones de servicio no solicitado. A consecuencia del control de servicio distribuido, el resultado de la ejecución de la lógica de servicio es transferido entre dos SCF;
- d) ejerce de interfaz e interactúa con las SDF para la captura de datos y manipulación de datos seguros;
- e) provee un punto de interconexión a la red con fines de conexión entre redes, ocultando de hecho la estructura específica de la red;
- f) ejerce de interfaz e interactúa con la SRF para interacciones relacionadas con la llamada para lo cual indica al SRF el guión de Interacción de usuario a ejecutar, proveyendo al SRF la información adicional que solicita durante la ejecución del guión de Interacción de usuario, y esperando a que finalice la ejecución del guión de Interacción de usuario;
- g) ejerce de interfaz e interactúa con las SRF para interacciones no relacionadas con la llamada controlando la disponibilidad de recursos en el SRF, pidiendo el control de algunos recursos de SRF fuera del contexto de una llamada;
- h) provee mecanismos de seguridad, con fines de conexión entre redes, para permitir una transferencia segura de la información a través del límite entre redes.

2.1.2.5 Función de datos de servicio (SDF, *service data function*); función SD

La SDF contiene datos de usuario y de red para acceso en tiempo real por la SCF en la ejecución de un servicio provisto por la RI:

- a) ejerce de interfaz e interactúa con las SCF para la manipulación y adquisición seguras de datos por medio de peticiones simples a la base de datos de guiones de gestión de datos;
- b) ejerce de interfaz e interactúa con otras SDF según se le requiera, permitiendo ocultar el lugar de la red en el cual se encuentran los datos. Este conocimiento se puede utilizar para obtener transparencia de la distribución de datos (por ejemplo, a la SCF);
- c) provee mecanismos de seguridad con fines de interconexión entre redes, a fin de posibilitar la transferencia segura de la información a través del límite entre redes;
- d) ejerce de interfaz e interactúa con otras SDF, permitiendo copiar datos junto con los derechos de acceso a dichos datos;
- e) provee autenticación y prestaciones de control de acceso a fin de proveer un acceso seguro a los datos de servicio;
- f) facilita la cooperación a nivel de gestión de tráfico, para evitar o resolver situaciones de congestión en la adquisición de datos;
- g) provee un soporte de datos para servicios de seguridad. Este soporte de datos puede ser utilizado por la misma SDF para una gestión segura de los datos;
- h) facilita la cooperación de un mecanismo robusto de recuperación para la copia de datos (por ejemplo, en los casos en que el SDF no esté disponible);
- i) provee guiones de acceso de datos (métodos) que pueden ser invocados por la SCF a fin de simplificar la transferencia de la información a través de la interfaz SCF-SDF. Este tipo de guiones de acceso de datos provee una manipulación simplificada de los datos entrantes. El SCF sigue proveyendo funciones específicas al servicio de lógica de procesamiento y control de llamadas de comando en la SSF.

NOTA – La SDF contiene datos relacionados con la prestación o explotación de servicios provistos por la RI. Por tanto, no necesariamente abarca datos proporcionados por terceros tal como información de crédito, pero puede proporcionar acceso a estos datos.

2.1.2.6 Función recursos especializados (SRF, *specialized resources function*); función SR

La SRF proporciona los recursos especializados necesarios para la ejecución de servicios provistos por la RI (por ejemplo, receptores de dígitos, anuncios, puentes de conferencia, etc.). Esta función:

- a) ejerce de interfaz e interactúa con la SCF y la SSF (y con la CCF);
- b) puede contener la lógica y la capacidad de procesamiento para recibir/enviar y convertir información recibida de usuarios;
- c) puede contener funcionalidad similar a la CCF para gestionar conexiones portadoras a los recursos especializados.

2.1.2.7 Función de recurso especializado (SRF) de reconocimiento automático del habla (ASR, *automatic speech recognition*)

El recurso ASR permite al usuario de servicios de RI ingresar instrucciones y datos usando la voz. El ASR puede tanto ser independiente como dependiente del hablante. En caso de un ASR dependiente del hablante, se debe proveer un mecanismo que permita al usuario gestionar directamente sus plantillas de voz utilizadas para reconocer instrucciones y datos. Un mecanismo de esta naturaleza deberá permitir al usuario examinar, actualizar, eliminar e insertar:

- las plantillas de voz; o bien

- las correspondencias entre las plantillas y el formato interno de la SRF de voz reconocida (por ejemplo, entre un nombre registrado con voz humana y la cadena de caracteres ASCII correspondiente).

Este mecanismo podría tanto ser controlado por la SCF, como implementado directamente por la SRF sin intervención de la SCF. En este último caso, la SRF informaría a la SCF del resultado de la operación si es que la SCF lo ha pedido. El recurso ASR básico debe permitir el reconocimiento de palabras aisladas (por ejemplo, los diez dígitos y cierto número de instrucciones elementales tales como "sí" y "no", pronunciados en al menos el idioma del proveedor de red local) de manera que sea independiente del hablante y se entregue por la RTPC.

Teniendo en cuenta que también podría resultar útil una ASR multilingüe, se reconoce que el SRF debería tratar la indicación del idioma solicitado para las entradas de viva voz de la misma manera descrita previamente para la generación de anuncios.

2.1.2.8 Función de recursos especializados de síntesis de texto hacia voz (TTS, *text-to-speech*)

La SRF puede contar con una función de texto a palabra. Esta funcionalidad consta de dos funciones lógicas. La primera función convierte el texto ingresado en una representación fonético-prosódica. La segunda función genera la señal vocal sintetizada, el procesamiento y la conexión de los elementos de voz.

2.1.2.9 Función de acceso inteligente (IAF, *intelligent access function*); función IA

La función de acceso inteligente (IAF) provee el acceso entre la SCF de una red estructurada como RI y una entidad que no es una red estructurada como RI. Esta última entidad puede consistir en otras redes o clientes (redes privadas, bases de datos simples utilizadas por ejemplo en el servicio CCR, terminales y PABX (centralitas automáticas privadas). Esta función:

- a) provee acceso de y a la SCF de la red estructurada como una RI;
- b) pone en correspondencia la información entre la representación interna y la representación externa;
- c) reside en la entidad que no es una red estructurada como RI;
- d) provee capacidades de pasarela que incluyen seguridad y funcionalidad cortafuegos.

2.1.2.10 Función de servicio no relacionada con la llamada (CUSF, *call-unrelated service function*); función CUS

La CUSF es la función de servicios no relacionados con la llamada la cual, asociada a la CCF y la SSF, provee un conjunto de funciones de servicio no relacionadas con la llamada que se necesitan para la interacción fuera del canal con una SCUAF. Provee también el conjunto de funciones requeridas para la interacción entre la SCUAF y una SCF. Esta función:

- a) extiende la lógica de la CCF para que incluya el reconocimiento de actuadores de control de servicio y para que interactúe con la SCF;
- b) gestiona la señalización entre la CCF y la SCF;
- c) modifica las funciones de procesamiento de conexión/asociación (en la CCF) según sea necesario para procesar requerimientos de utilización de servicios provistos por la RI bajo el control de la SCF;
- d) modifica las funciones de procesamiento de interacciones no relacionadas con la llamada (en la CUSF) según se requiera para procesar peticiones de utilización de servicios provistos por la RI bajo el control de la SCF;
- e) soporta la interacción con usuario no relacionada con la llamada, que puede ser iniciada por el usuario o por la SCF;
- f) establece la interfaz a la SSF para manejar las interacciones relativas a la llamada.

2.1.2.11 Función de agente de usuario de control de servicio (SCUAF, *service control user agent function*); función SCUA

La SCUAF es la función de agente del usuario de control de servicio que provee el acceso a los usuarios. Es la interfaz entre un usuario y la función CUSF. Esta función:

- a) provee el acceso al usuario, para lo cual interactúa con el mismo para establecer, mantener y liberar -según se requiera- una instancia de servicio no relativo a la llamada;
- b) accede a las capacidades de provisión de servicio de la función de control de llamada (CCF), utilizando peticiones de servicio (por ejemplo establecimiento, registro de posición) para establecer, manipular y liberar una asociación o una instancia de servicio;
- c) recibe indicaciones relativas a servicios no relativos a llamadas de la CCF y los retransmite al usuario según sea necesario;
- d) mantiene la información de estado del servicio según la percibe esta entidad funcional.

NOTA – En RI CS-2 no se define si la SCUAF abstrae una nueva FE para servicios suplementarios asociados a la llamada. Asimismo, RI CS-2 no define la relación a usar para crear un modelo de la interacción con el usuario relacionada con la llamada (la relación existente entre la CCAF y la CCF o una relación explícita entre algunas FE).

2.1.2.12 Función de gestión de servicio (SMF, *service management function*); función SM

La SMF es la función de gestión de servicio. En esta cláusula se describen varias funcionalidades SMF de la RI. Estas funciones se pueden agrupar en cinco categorías:

- 1) funciones de instalación de servicio (*service deployment functions*);
 - 2) funciones de provisión de servicio (*service provisioning functions*);
 - 3) funciones de control de operación de servicio (*service operation control functions*);
 - 4) funciones de facturación (*billing functions*);
 - 5) funciones de control de servicio (*service monitoring functions*).
- Las funciones de instalación de servicio incluyen:
 - Asignación de guiones de servicio:
Esta subfunción cursa los guiones de servicio, determina a qué parte de la red son pertinentes los guiones de servicio, y gestiona los elementos de red pertinentes.
 - Asignación de datos genéricos del servicio:
Esta subfunción cursa los datos genéricos del servicio, determina a qué parte de la red los datos genéricos del servicio son pertinentes y gestiona los elementos de red pertinentes.
 - Introducción y asignación de datos de encaminamiento de señalización:
Esta subfunción cursa los datos de encaminamiento de señalización, determina a qué parte de la red los datos de encaminamiento de señalización son pertinentes y gestiona los elementos de red pertinentes. Telecarga los datos de encaminamiento de señalización en la red SS7 y determina qué elementos de la red SS7 son pertinentes a la asignación de los datos de encaminamiento de señalización.
 - Introducción y asignación de datos de activación:
Esta subfunción cursa los datos de activación, determina a qué parte de la red son pertinentes los datos de activación y gestiona los elementos de red pertinentes. Telecarga los datos de activación en la RTPC.

- Introducción y asignación de datos de recursos especializados:
Esta subfunción cursa los datos de recursos especializados, determina a qué parte de la red son pertinentes los datos de recursos especializados y gestiona los elementos de red pertinentes.
- Pruebas de servicio:
Esta subfunción reúne el soporte lógico de servicio de la función entorno de creación de servicio (Service Creation Environment Function) a ser cargado en una red RI autónoma a fin de probar el servicio que se acaba de desarrollar. La función ingresa los datos específicos al servicio y al usuario. Realiza operaciones de prueba relativas a la gestión.
- Las funciones de provisión de servicio incluyen:
 - Introducción y asignación de datos específicos al cliente:
Esta subfunción reúne datos específicos al abonado al servicio y los administra en bases de datos de abonados y en bases de datos de contratos. La función traduce los datos del servicio y del abonado a datos que son específicos de la red. Esta subfunción determina a qué parte de la red estos datos son pertinentes, y gestiona los elementos de red pertinentes.
- Las funciones de control de operación de servicio incluyen:
 - Mantenimiento del servicio:
El mantenimiento del servicio incluye la siguiente funcionalidad:
 - Mantenimiento del soporte lógico:
El mantenimiento del soporte lógico consiste en modificar la lógica de servicio (la modificación de la lógica de servicio es una función de la SCEF). La introducción de un guión modificado en la red estructurada como RI se realiza durante la instalación del servicio.
 - Actualización de datos genéricos del servicio:
Esta subfunción cursa los datos genéricos del servicio, determina a qué parte de la red son pertinentes los datos genéricos del servicio y gestiona los elementos de red pertinentes.
 - Actualización de datos específicos al usuario:
Esta subfunción provee las funciones de control para datos específicos al abonado al servicio y administración de dichas bases de datos de abonados y bases de datos de contratos. Esta subfunción determina a qué parte de la red son pertinentes los datos, y gestiona los elementos de red pertinentes.
 - Actualización de datos de encaminamiento de señalización:
Esta subfunción provee las funciones de control para los datos de encaminamiento de señalización, determina a qué parte de la red son pertinentes los datos de encaminamiento de señalización y gestiona los elementos de red pertinentes. Descarga los datos de encaminamiento de señalización a la red SS7 y determina los elementos de la red SS7 pertinentes a la asignación de los datos de encaminamiento de señalización.
 - Actualización de datos de activación:
Esta subfunción provee las funciones de control para los datos de activación, determina a qué parte de la red son pertinentes los datos de administración, y gestiona los elementos de red pertinentes. Descarga los datos de activación a la RTPC.
 - Actualización de datos de recursos especializados:
Esta subfunción provee las funciones de control para los datos de recursos especializados, determina a qué parte de la red son pertinentes los datos de recursos especializados, y gestiona los elementos de red pertinentes.

- Ajuste de la SMAF

La interfaz a la SMF del abonado al servicio/operador de la red es provista por la SMAF. La interfaz al abonado al servicio y al operador de la red debe ser compatibilizada con los ajustes operados en sus datos. Por ejemplo, un abonado al servicio que ha cambiado de tipo de periférico (datos específicos del abonado: teléfono DTMF a terminal VTX). Este cambio de periférico puede también ser causa de una modificación de las opciones de menú.
- Reconfiguración del servicio:

Esta actividad consiste en volver a asignar los guiones de servicio, los datos genéricos de servicio y los datos específicos de abonado. Por ejemplo, la razón de la reconfiguración del servicio puede ser una modificación de la configuración de la red o una mejora en la prestación de los servicios.
- (Des)activación del servicio

Esta actividad le da al operador de la red la posibilidad de (des)activar (parte de) un servicio en forma temporaria. Se aplica por ejemplo a las actividades de mantenimiento de un servicio de votación telefónica que sólo se usa durante determinados periodos de tiempo.
- Desmantelamiento de un servicio:

Hará que un servicio deje de funcionar.
- Seguridad:

Cabe distinguir entre dos tipos de seguridad en la SMF: control de acceso y control de datos. El control de acceso se ocupa de la identificación, autenticación y autorización (control de instrucciones) tanto del abonado al servicio como del operador de la red. El control de datos se ocupa del control de la entrada de datos tanto por el abonado al servicio como por el operador de la red.
- Las funciones de facturación comprenden:
 - Generar y almacenar registros de tasación:

Esta subfunción controla la utilización del servicio. Esta función lleva un informe de los registros de llamada.
 - Reunir registros de tasación:

Esta subfunción reúne los registros de tasación y los registros detallados de gestión. Luego los uniformiza y los correlaciona. Esta función lleva un informe de los registros de llamada.
 - Modificación de tarifas:

Esta subfunción determina la estructura tarifaria y la tarifa para un servicio recién elaborado, o las sustituye por una existente.
- El control del servicio incluye:
 - Iniciar mediciones y reunir datos de medición:

Esta subfunción controla la utilización y calidad de funcionamiento del servicio. Controla también el desempeño de la red. Por lo tanto, necesita los resultados de mediciones de las partes subyacentes: la función de gestión de SS7 y la función de gestión de red.
 - Análisis y generación de informes de datos de medición:

Esta subfunción analiza la utilización y calidad de funcionamiento del servicio. También analiza los resultados de la función de datos de medición de iniciación y recolección.

- Recepción de información de datos de control de averías:

Esta subfunción se activa al recibir datos de control de averías provenientes de los elementos de red. Se determinarán las consecuencias y repercusiones sobre la calidad de funcionamiento del servicio y se tomarán las medidas necesarias.

2.1.3 Relaciones entre entidades funcionales

Los servicios se soportan por medio de las acciones conjuntas de un grupo de entidades funcionales. Esta cooperación hace necesario establecer relaciones de comunicación.

- Se dice que cada par de entidades funcionales que se comunican, dentro del modelo funcional de un servicio específico, están en relación.
- Cada interacción entre un par de entidades funcionales en comunicación se denomina como un flujo de información o una llamada a API (este último caso corresponde a la utilización de UML para describir el punto de referencia). La relación entre un par cualquiera de entidades funcionales es el conjunto completo de flujos de información o de llamadas a API entre ellas. En este último caso, se conoce como el API para dicho punto de referencia.
- Si dos entidades funcionales en comunicación están situadas en equipos separados físicamente, los flujos de información o las llamadas a API entre ellas definen los requisitos de transferencia de información para un protocolo de señalización entre los equipos.
- Diferentes pares de entidades funcionales en comunicación pueden tener relaciones de tipos diferentes. El tipo de una relación se caracteriza por el conjunto de flujos de información o de llamadas a API entre dos entidades funcionales. Por ejemplo, se dice que las relaciones entre las entidades funcionales FE1 y FE2 y entre las entidades funcionales FE3 y FE4 son del mismo tipo, si incluyen el mismo conjunto de flujos de información o de llamadas a API.
- Se pueden asignar identificadores de tipo a las relaciones (por ejemplo r1, r2, r3, etc.), que identifican unívocamente conjuntos específicos de flujos de información dentro del modelo funcional del servicio. Un mismo tipo de relación puede ocurrir más de una vez en el modelo funcional.

2.1.4 Diseño del modelo funcional

A base de las definiciones anteriores, se deduce el modelo funcional de un servicio concreto utilizando los siguientes criterios y reglas:

- Las entidades funcionales apropiadas se eligen en función del conocimiento de la diversidad de realizaciones previstas de la red. Se tendrán en cuenta todas las distribuciones razonables de funciones, dejando por tanto a criterio de la administración la manera de ofrecer el servicio.
- Los tipos de relaciones se asignan inicialmente en base a una evaluación de la posible naturaleza de las interacciones entre cada par de entidades funcionales. Pueden ser necesarias revisiones del modelo inicial a la luz de una definición más detallada de las acciones de la entidad funcional, los flujos de información o llamadas a API y las posibilidades de ubicación física de las entidades funcionales.
- El modelo para alguno de los servicios, puede requerir que una unidad funcional sea iterada varias veces (por ejemplo, funciones en cascada). El modelo funcional deberá describir únicamente las iteraciones hasta el momento en el que, por medio de otra iteración, no se encuentren nuevas combinaciones entre relaciones externas con entidades funcionales. Por tanto, una única entidad funcional puede representar a varias entidades físicas en cascada que proporcionen las mismas funciones.

El modelo funcional unificado está destinado a captar aspectos de tantas arquitecturas de red como sea posible. En la RDSI, aunque muchas de las FE del modelo están normalmente cubiertas en implementaciones físicas, la funcionalidad del servicio puede separarse entre las funciones

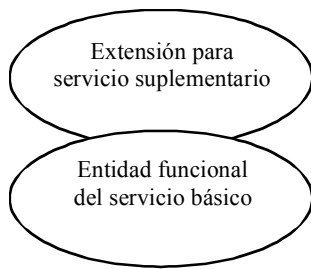
especificadas. En la red inteligente y en las IMT-2000, el modelo funcional y las FE son típicas del conjunto de capacidades de la RI y de las arquitecturas IMT-2000 (véase UIT-T Q.1711). En la RDSI-BA, el modelo funcional unificado identifica relaciones borde a borde entre funciones de control de llamada servidoras y relaciones enlace a enlace utilizando funciones de control de llamada de tránsito. Las configuraciones punto a multipunto son atendidas identificando instancias adicionales de entidades funcionales similares y las correspondientes relaciones. Para las redes basadas en el IP, el modelo funcional proporciona un enlace a los objetos gestionados que puede considerarse existen dentro de las FE y, como tales, pueden ser direccionados y accedidos a través de esas FE. El modelo funcional también permite acceder a funciones que proveen la correspondencia de protocolos que pueda resultar necesaria entre las redes RI e IP bajo la forma de funciones de pasarela.

2.1.5 Relación entre los modelos de los servicios básico y suplementario

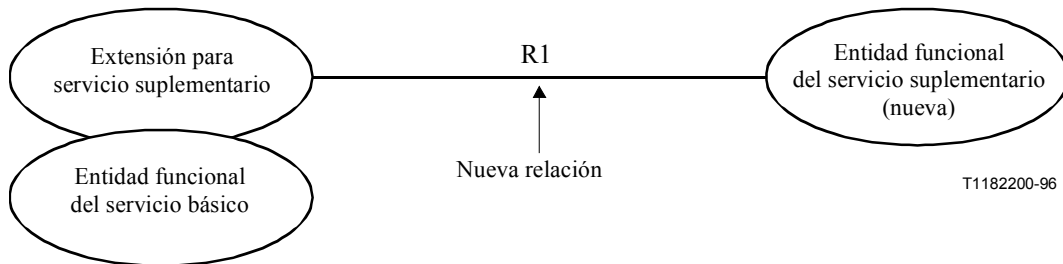
El modelo funcional de un servicio suplementario estará relacionado con un modelo de servicio básico. Se deben seguir las siguientes directrices para decidir si las funciones asociadas con un servicio suplementario deben estar coubicadas con entidades funcionales del servicio básico existentes o como entidades funcionales nuevas que no están coubicadas:

- Una agrupación de funciones dentro del modelo de un servicio suplementario debe estar coubicada con una entidad funcional del servicio básico (por ejemplo, véase la figura 5) en caso de que modifique un objeto controlado por el servicio básico (por ejemplo, conexión de llamada).
- Una entidad funcional que no esté coubicada con una entidad funcional del servicio básico no requerirá, típicamente, información detallada de estado de la llamada/conexión. Una entidad funcional independiente puede ser también caracterizada por una relación transaccional con una entidad funcional del servicio suplementario, coubicada con una entidad funcional del servicio básico (por ejemplo, para la traducción del número para la entidad funcional del servicio básico).
- Se define una relación entre las entidades funcionales para el servicio suplementario si, y sólo si, las entidades funcionales del servicio suplementario necesitan comunicarse y esas necesidades de comunicación no son satisfechas por los flujos de información definidos dentro del servicio básico.
- La relación entre el modelo para un servicio suplementario y para un servicio básico puede obtenerse comparando los modelos. Se aclara así cómo las entidades funcionales del modelo del servicio suplementario necesitan tener en cuenta todos los escenarios definidos en el paso 6 para el modelo del servicio básico y para el modelo del servicio suplementario.

La figura 4 ilustra estas relaciones.



a) Funciones adicionales para el servicio suplementario proporcionadas como extensión a una entidad funcional de un servicio básico



b) Funciones adicionales para un servicio suplementario proporcionadas como una entidad funcional no coubicada separada

Figura 4/Q.65 – Formas alternativas de añadir funciones de servicio suplementario al modelo funcional de servicio básico

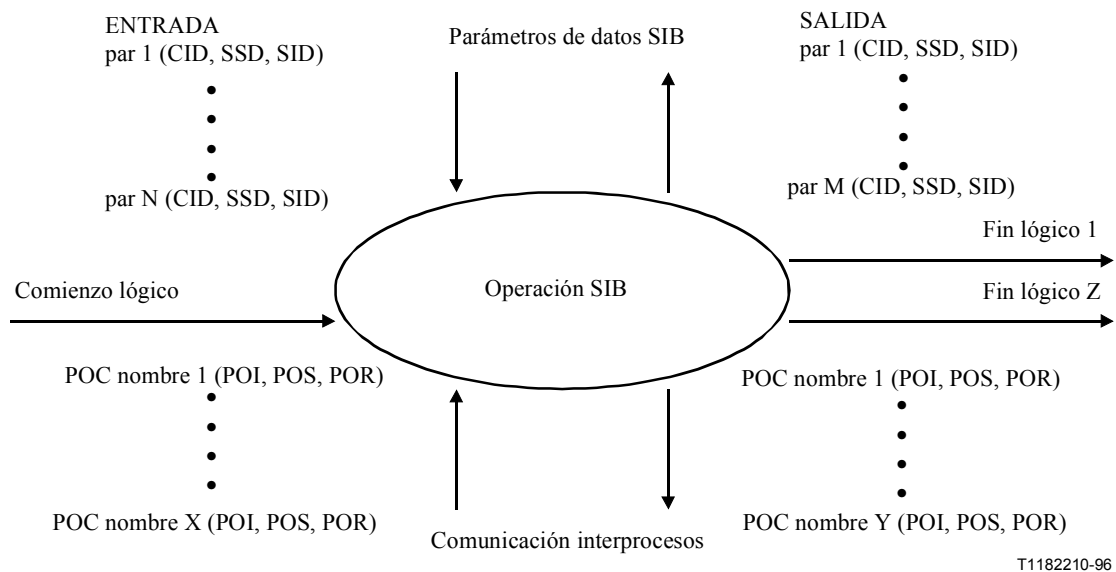


Figura 5/Q.65 – Representación gráfica de una operación SIB

2.2 Paso 2 (opcional) – Descripción SIB de las características de los servicios (véase en 3.1, el paso 2 alternativo que utiliza técnicas orientadas a objetos)

Bloques de construcción independientes del servicio

Un SIB es una capacidad a nivel de red reutilizable normalizada que reside en el plano funcional global y se utiliza para crear características de servicios (véase la figura 1). Los SIB son de

naturaleza global y su realización detallada no se considera a este nivel, pero pueden hallarse en el plano funcional distribuido (DFP, *distributed functional plane*) y en el plano físico. Los SIB son reutilizables y pueden combinarse para realizar servicios descritos en el plano de servicios. La capacidad ofrecida dentro del SIB es indicada por el conjunto de operaciones que pueden ser invocadas en los SIB. El conjunto de operaciones que el SIB ofrece constituye la interfaz SIB. Cada operación define una función que puede efectuarse en relación con la capacidad del SIB. Los SIB complejos, por ejemplo, los que modelan actividades persistentes se definen por medio de varias operaciones que permiten el control de la actividad realizada por el SIB. Los SIB son por definición independientes del servicio concreto y de la tecnología para la cual o con arreglo a la cual serán realizados.

Características de un SIB

- Los SIB son bloques de construcción monolíticos (su implementación detallada está oculta) que el diseñador de servicios utilizará para desarrollar nuevos servicios.
- Todas las características de servicio (SF, *service features*) se describen mediante un SIB o una combinación de SIB.
- Todas las SF pueden definirse mediante un número finito de SIB.
- Un SIB define una actividad completa.
- Los SIB son realizados en el DFP por acciones de entidades funcionales que pueden residir en una o más entidades funcionales (FE).
- Una operación SIB tiene un punto de comienzo lógico y uno o más puntos de fin lógicos. Los datos requeridos por cada operación SIB son definidos por parámetros de datos soporte de operaciones SIB y parámetros de datos de instancia de llamada.
- Los SIB son de carácter global, y sus ubicaciones no necesitan ser consideradas, ya que el conjunto de la red se considera una sola entidad en el GFP (Global Functional Plane, Plano funcional global).
- Los SIB son reutilizables. Se utilizan sin modificación para otros servicios.

Esta subcláusula pretende ser una introducción a la utilización de bloques de construcción independientes del servicio (SIB, *service independent building blocks*) en la metodología funcional unificada (UFM) para descripciones de servicios. El concepto de SIB ha sido adoptado para tratar las necesidades de creación de servicios, así como para introducir bloques reutilizables de flujos, SDL y FEA que puedan ser catalogados. La adición del concepto de SIB se introduce para simplificar los trabajos de la definición de servicios de la etapa 2. Se crearán los nuevos SIB que se necesiten, y se mantendrá una biblioteca de SIB como referencia para la creación y definición de servicios.

NOTA – El uso de SIB para descripciones de la etapa 2 no se limita únicamente a servicios soportados por la RI.

El planteamiento del modelo conceptual de RI de definir y utilizar SIB ofrece la base para un método flexible de creación y descripción de servicios, así como un método para referenciar flujos predefinidos y el SDL. El concepto de SIB (al que se hace referencia en UIT-T Q.1213/Q.1214; (1995)) se ha elegido como una adición clave a la metodología de la etapa 2 descrita en esta Recomendación.

Existe terreno común para la metodología unificada en el sentido de que los SIB se descomponen en FEA, SDL, y flujos de información. El atractivo del método estriba en el potencial de reutilización (los SIB se hacen corresponder a acciones de entidad funcional predefinidas descritas en flujos de información y SDL predefinidos). No puede demostrarse en absoluto que los SIB puedan combinarse en todas las formas posibles para crear nuevos servicios, pero se están considerando varios métodos (por ejemplo, extensión del proceso SDL de incluir la combinación de SIB). La verificación de los SIB existentes (es decir, que cada uno está correctamente definido) es posible analizando los SDL que los definen. La verificación de las combinaciones de SIB es posible

analizando los SDL creados cuando los SIB se combinan para describir características de servicio y capacidades de red.

Hay necesidad de nuevos SIB, así como de la elaboración de reglas de combinación y reglas de aplicación. Los SIB deben crearse "de arriba a abajo" a partir de los servicios proyectados, pero también de "abajo a arriba" a partir de la experiencia con la metodología de descripción de servicios. De este modo, la metodología puede tratar las necesidades actuales, pero también evolucionar para que incluya mejoras y nuevas tecnologías. Con los continuados esfuerzos en la definición de SIB y reglas de aplicación, los SIB podrían convertirse en la primera herramienta para la descripción de servicios y creación de servicios de la etapa 2.

2.2.1 Definiciones de SIB

Terminología:

BCP	Proceso de llamada básica (<i>basic call process</i>)
CID	Datos de instancia de llamada (<i>call instance data</i>)
GSL	Lógica de servicio global (<i>global service logic</i>)
HLSIB	SIB de alto nivel (<i>high level SIB</i>)
POC	Punto de control (<i>point of control</i>)
POI	Punto de iniciación (<i>point of initiation</i>)
POR	Punto de retorno (<i>point of return</i>)
POS	Punto de sincronización (<i>point of synchronization</i>)
SID	Datos de instancia del servicio (<i>service instance data</i>)
SSD	Datos de soporte del servicio (<i>service support data</i>)

2.2.1.1 bloques de construcción independientes del servicio de proceso de llamada básica

El proceso de llamada básica (BCP) se encarga de proporcionar conectividad de llamada básica entre partes en la red. El BCP puede verse como un proceso de servicio especializado, que proporciona capacidades de procesamiento de eventos de la llamada básica, así como un SIB especializado, que proporciona un conjunto de operaciones SIB, tales como:

- conexión de llamadas, con disposición apropiada;
- desconexión de llamadas, con disposición apropiada;
- retención de CID para posterior procesamiento de esa instancia de llamada.

2.2.1.2 bloques de construcción independientes del servicio de alto nivel (HLSIB, *high level SIB*)

Los SIB de alto nivel (HLSIB) son, como los SIB normales, una parte reutilizable de una característica de servicio, y están compuestos por operaciones SIB y otros HLSIB que pueden ejecutarse secuencialmente. Los HLSIB tienen las siguientes características adicionales:

- Los HLSIB pueden componerse de otros HLSIB y de operaciones SIB solamente.
- Un cierto HLSIB no puede utilizarse como un componente dentro del mismo HLSIB, es decir, no es posible ningún uso recursivo.
- El nivel más bajo de HLSIB contiene operaciones SIB solamente, es decir, ningún otro detalle es visible en el GFP.
- Uno de los (HL)SIB dentro de un HLSIB es el primero en ser ejecutado; por tanto, los HLSIB tienen sólo un punto de entrada (comienzo lógico), al igual que con SIB normales.

Sin embargo, como ocurre también con los SIB normales, los HLSIB pueden tener uno o más puntos de salida (fines lógicos).

2.2.1.3 Lógica de servicio global

La GSL puede definirse como el "pegamento" que define el orden en el que se encadenarán entre sí las operaciones SIB para construir procesos de servicio a fin de realizar características de servicio. Cada instancia de la lógica de servicio global es (potencialmente) única para cada llamada individual, pero utiliza elementos comunes, que comprenden específicamente:

- puntos de interacción (POI, POS y POR) de procesos de servicio, incluido el proceso de servicio BCP;
- operaciones SIB;
- conexiones lógicas entre operaciones SIB, y entre operaciones SIB y puntos de interacción de procesos de servicio;
- parámetros de datos de entrada y salida, datos soporte del servicio y datos de instancia de llamada definidos para cada SIB.

Basándose en la funcionalidad de estos elementos comunes, la lógica de servicio global "encadenará" estos elementos para proporcionar un servicio específico.

2.2.2 Parámetros de datos SIB

Por definición, los SIB son independientes del servicio/SF que suelen representar. No tienen ningún conocimiento sobre otros SIB que se utilizan para describir la característica de servicio.

A fin de describir características de servicio con estos SIB genéricos, se necesitan algunos elementos de dependencia del servicio.

La dependencia del servicio puede describirse utilizando parámetros de datos que permitan a un SIB ser ajustado para efectuar la funcionalidad deseada. Los parámetros de datos se especifican independientemente para cada SIB y se hacen disponibles al SIB mediante lógica de servicio global.

Los parámetros de datos constan de parámetros de entrada y de salida. Se requieren dos tipos generales de parámetros de datos para cada operación SIB, parámetros dinámicos denominados datos de instancia de llamada (CID) y parámetros estáticos denominados datos soporte de servicio (SSD) y datos de instancia de servicio (SID).

La diferenciación entre los parámetros SIB formales y los parámetros reales introduce más flexibilidad al asignar el tipo de datos de parámetro SIB. Los parámetros SIB formales son los parámetros que se utilizarán para descripciones de SIB en esta Recomendación. Los parámetros SIB reales sólo aparecen en instancias SIB en lógica de servicio global específica (GSL).

2.2.2.1 datos de instancia de llamada (CID, *call instance data*)

Los datos de instancia de llamada definen parámetros dinámicos cuyo valor cambiará con cada instancia de llamada. Se utilizan para dar detalles de abonado específicos tales como información de línea llamante o llamada. Estos datos pueden ser:

- hechos disponibles desde el BCP (por ejemplo, identificación de la línea llamante);
- generados por una operación SIB (por ejemplo, un número traducido); o
- introducidos por el abonado (por ejemplo, un número marcado o un código PIN).

2.2.2.2 datos de soporte del servicio (SSD, *service support data*)

Los datos de soporte del servicio definen parámetros de datos requeridos por una operación SIB que son específicos para la descripción de características del servicio. Cuando se incluye una operación en la GSL de una descripción del servicio, la GSL especificará los valores SSD para el SIB. SSD consta de parámetros fijos. Estos son parámetros de datos cuyos valores son fijos para todas las

instancias de llamada. Por ejemplo, el "indicador de fichero" SSD para el SIB traducción necesita especificarse unívocamente para cada aparición de SIB en una característica de servicio dada. El valor SSD de "indicador de fichero" se dice entonces que es fijo, ya que su valor viene determinado por la descripción del servicio/SF, y no por la instancia de llamada.

Si un servicio/SF se describe utilizando múltiples apariciones del mismo SIB, se definen entonces parámetros SSD fijos unívocamente para cada aparición.

2.2.2.3 datos de instancia del servicio (SID, *service instance data*)

Datos de instancia del servicio definen datos relacionados con un perfil del abonado del servicio, que existe antes de que se invoque el servicio y pueden modificarse y actualizarse como resultado de la actividad de procesamiento del servicio. Este tipo de datos puede leerse con la ejecución del servicio y almacenarse para ser utilizado en posteriores invocaciones del servicio.

2.2.3 Convenios de modelado de SIB

Representación gráfica

Se utiliza una representación gráfica para describir cada operación efectuada por un SIB. Se ilustra en la figura 6. Cada operación SIB se caracteriza por tener parámetros de entrada y de salida, un flujo lógico de entrada, y uno o más flujos lógicos de salida. Estos flujos lógicos se muestran mediante las flechas de trazo continuo situadas a la izquierda y derecha del diagrama. Cada flujo lógico se especifica encima de cada flecha. Los parámetros de entrada y de salida se identifican por las flechas de trazo discontinuo encima del diagrama y se especifican al lado de la flecha de trazo discontinuo. Para los parámetros de entrada y los de salida, el tipo de SSD, SID y CID se declara al lado de los parámetros respectivos. Análogamente, los POC se especifican debajo del diagrama.

Los SIB pueden definirse con diversos grados de granularidad. Por composición, los SIB pueden definirse a partir de SIB más pequeños, formando un SIB de alto nivel (HLSIB). Inversamente, por descomposición se consigue la división de la granularidad de un HLSIB en bloques más pequeños que pueden ser reutilizados. La figura 6 muestra varias capas de granularidad para los HLSIB. Puede ser conveniente catalogar ciertos HLSIB, así como combinaciones de SIB simples a fin de favorecer la reutilización de funcionalidad más compleja.

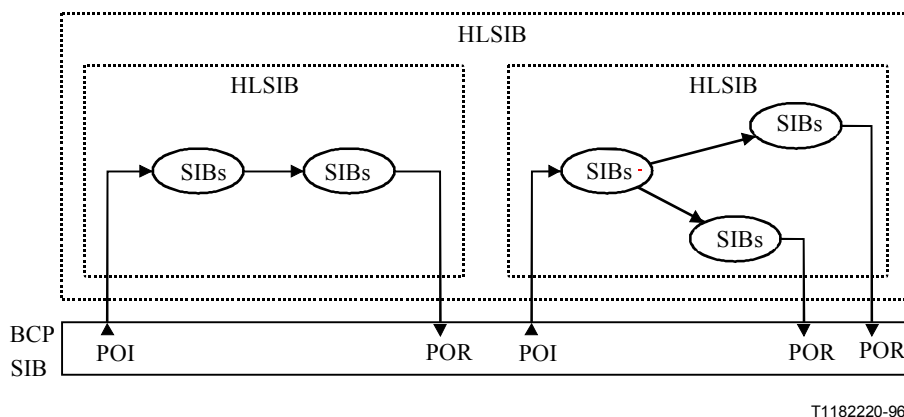
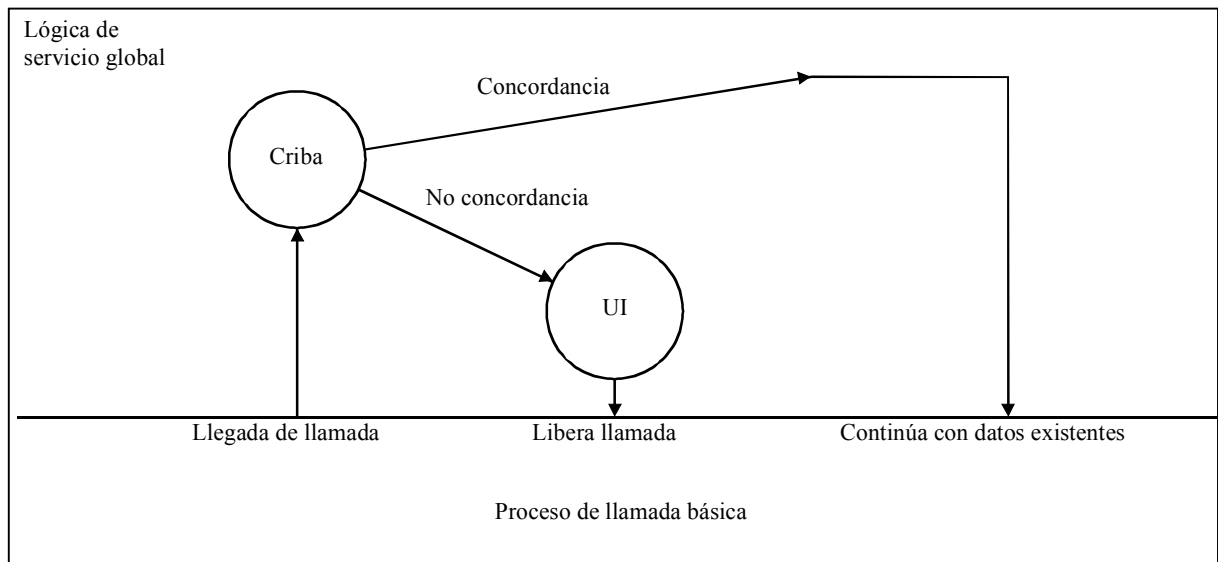


Figura 6/Q.65 – Combinación de SIB/HLSIB (ejemplo genérico)

La figura 7 presenta un ejemplo de un diagrama de lógica de servicio global. El diagrama de lógica de servicio global, que muestra el SIB, o los SIB en combinación, efectuando una determinada función de servicio es un "puente" perfecto desde una descripción de la etapa 1 hasta la descripción de la etapa 2. El diagrama describe el servicio o la capacidad de red en forma taquigráfica que presenta una gran cantidad de información con un medio gráfico relativamente simple. Las

definiciones contenidas en el diagrama, sin embargo, contienen todo el SDL, flujos de información y FEA necesarias para completar la descripción de la etapa 2.

En la figura 7 se muestra una posible definición del servicio de cribado de terminación. Desde el punto POI de llegada de célula, se utiliza el SIB CRIBA (SCREEN) para determinar si el usuario llamante está en la lista de los usuarios a los que se permite terminar una llamada en el destino. Si está en la lista, se permite la llamada, y el BCP continúa el tratamiento de la llamada con los datos existentes. Si el usuario llamante no está en la lista, el SIB INTERACCIÓN USUARIO (USER INTERACTION) se utiliza para entregar un mensaje de desconexión apropiado al llamante, en cuyo momento el BCP libera la llamada.



T1182230-96

Figura 7/Q.65 – Servicio de cribado de terminación de GFP

2.2.4 Modelado SIB de características de servicio

Se alienta a que los SIB y los HLSIB se utilicen para describir características de servicio y capacidades de red, ya que favorece la reutilización de SDL, flujos de información, y FEA que están ya definidos. El diagrama de lógica de servicio global, que muestra el SIB, o SIB en combinación, efectuando una función de servicio determinada es un medio eficiente y eficaz de proveer la información SIB a la descripción de la etapa 2.

Las que siguen son directrices para el uso de SIB en el método de la etapa 2:

- Proporcionar diagramas GSL de características de servicio que incluyan POI(s) procedentes de y POR(s) destinados al SIB BCP.
- Poner en correspondencia los SIB utilizados en la descripción con las FE del modelo funcional unificado. Además, proporcionar referencias para las definiciones de los SIB utilizados (por ejemplo, cláusula 5/Q.1213 (1995)). Esto puede hacerse fácilmente en forma tabular (véase 2.2.6).
- Proporcionar los parámetros de datos SIB necesarios para el servicio. Identificar valores de datos dentro del contexto del servicio.

Si todas las características de servicio pueden describirse completamente con SIB, deben proporcionarse referencias específicas para el SDL, flujos de información y FEA de soporte en las cláusulas apropiadas de la descripción de la etapa 2. No se necesita redacción explícita de estas cláusulas (siempre que las descripciones SIB hayan demostrado ser exactas. Estas "pruebas" para las

diversas combinaciones de SIB y los HLSIB vendrán con la experiencia; la biblioteca de SIB atenderá esta función). Si no hay SIB disponibles para describir una determinada característica, puede todavía utilizarse la metodología funcional unificada como ha sido siempre utilizada para derivar explícitamente el SDL, flujos de información y FEA. Si se identifican algunos SIB para describir parcialmente un servicio, se sugiere que los diagramas SDL y los flujos de información se muestren explícitamente con indicaciones en cuanto a qué subcláusulas de los flujos pertenecen las definiciones de SIB.

2.2.5 Lista de SIB disponibles

Bloques de construcción independientes del servicio CS-1R de RI (SIB):

- 1) algoritmo;
- 2) autenticación;
- 3) tarificación;
- 4) comparación;
- 5) distribución;
- 6) límite;
- 7) información de registro de llamada;
- 8) cola de espera;
- 9) criba;
- 10) gestión de datos de servicio;
- 11) notificación de situación;
- 12) traducción;
- 13) interacción de usuario.

En la serie de Recomendaciones Q.12x3 se definen otros SIB.

2.2.6 Correspondencia de SIB con FE

El cuadro 1 muestra la correspondencia SIB/FE.

Cuadro 1/Q.65 – Correspondencia SIB/FE

SIB	Entidades funcionales			
	SSF/CCF	SCF	SRF	SDF
Algoritmo		X		
Tarificación	X	X		
Comparación		X		
Distribución		X		
Límite	X	X		
Información de registro de llamada	X	X		X
Cola de espera	X	X	X	
Criba		X		X
Gestión de datos de servicio		X		
Notificación de situación	X	X		
Traducción		X		X
Interacción de usuarios	X	X	X	
Verificación		X		
Proceso de llamada básica	X	X		
Autenticación		X		X

2.3 Paso 3 – Diagramas de flujo de información (véase en 3.2 el paso 3 alternativo que utiliza técnicas orientadas a objetos)

2.3.1 Identificación de los flujos de información

La distribución de las funciones necesarias para suministrar un servicio, como lo define el modelo funcional, requiere interacciones entre entidades funcionales. A tal interacción se denomina "flujo de información" y tendrá un nombre descriptivo del propósito del flujo de información.

Se establecen diagramas de flujo que comprenden todos los flujos de información necesarios para los casos típicos de operaciones correctas del servicio. Para otros casos puede ser necesaria la creación de otros diagramas de flujo de información. La figura 8 ilustra la forma general de un diagrama de flujo de información para un servicio básico o suplementario.

Los diagramas de flujo de información de los servicios suplementarios no deben duplicar innecesariamente las descripciones de los flujos de información que forman parte de un servicio básico. Sin embargo, puede ocurrir que una descripción de servicio suplementario identifique requisitos adicionales de flujo de información entre las entidades funcionales de la representación del servicio básico, y ello debe ser descrito.

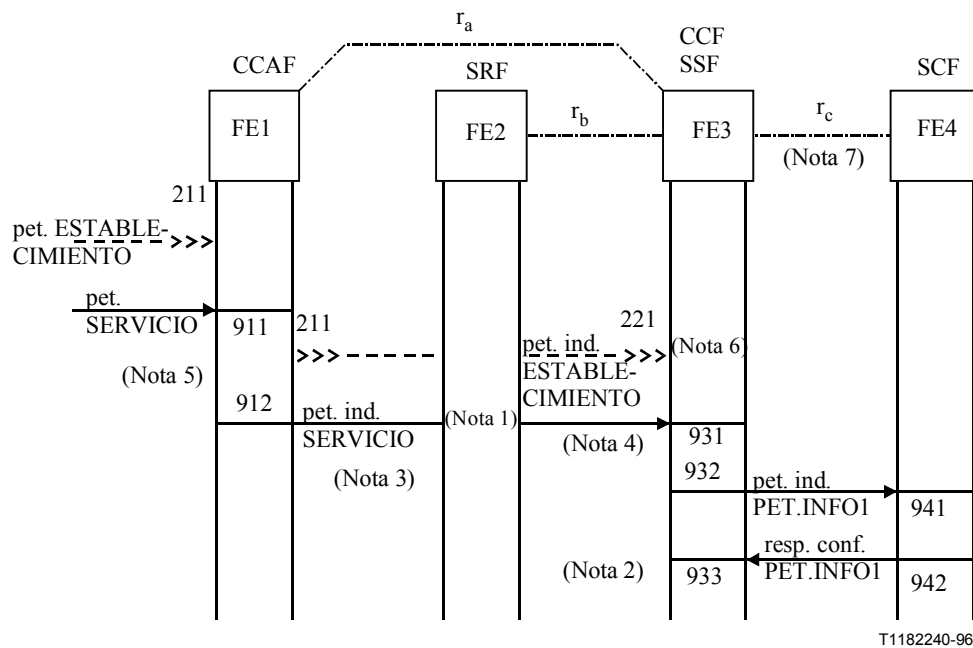


Figura 8/Q.65 – Ejemplo de diagrama de flujo de información

Notas relativas a la figura 8:

NOTA 1 – La recepción y emisión de entradas/salidas del usuario y flujos de información vienen indicadas por medio de líneas horizontales que cruzan las columnas de las entidades funcionales. La ausencia de línea indica que no hay ni recepción ni emisión.

NOTA 2 – Se asigna un número de referencia a cada punto en la secuencia completa en la que se presentan las acciones de las entidades funcionales.

NOTA 3 – Los flujos de información se representan por flechas, con el nombre del flujo de información encima y debajo de la flecha. El nombre descriptivo aparece en mayúsculas debajo de la flecha, y la etiqueta aparece encima de ella en minúsculas (por ejemplo, ind. pet.). Para los flujos de información sin confirmación y para la parte "petición" de los flujos de información con confirmación, se presenta la etiqueta "ind. pet." en letra minúscula encima de las flechas de flujo de información. Para la parte "confirmación" de los flujos de información con confirmación se utiliza la etiqueta "resp. conf.".

NOTA 4 – En la columna de una entidad funcional concreta:

- Las acciones que se indican debajo de una línea que representa la recepción de una entrada de usuario o flujo de información dependen de lo que se reciba (es decir, no se pueden llevar a cabo antes de que haya tenido lugar la recepción). Por tanto, por ejemplo, la acción 931 no se puede efectuar antes de que se haya recibido SERVICIO.
- Análogamente, las acciones que se indican encima de una línea, que representa la emisión de una salida de usuario o flujo de información, deben completarse antes de la emisión del flujo de información. Por tanto, no se puede emitir PET.INFO1 hasta que se hayan completado las dos acciones 931 y 932. No hay ninguna implicación en cuanto al orden de ejecución de las acciones 931 y 932.
- Las acciones que se indican debajo de una línea, que representa la emisión de una salida de usuario o flujo de información, no es necesario que se completen antes de la emisión (aunque en muchas realizaciones prácticas puede que sea así). No se restringe el orden relativo de la emisión y de la acción que sigue inmediatamente. Por tanto, acción 942 se puede ejecutar antes, después o en paralelo con la emisión de la parte "petición" del flujo de información PET.INFO1.

NOTA 5 – Las interacciones del servicio de la etapa 1 son las entradas y salidas del diagrama de flujo de información de la etapa 2. Las interacciones del servicio de la etapa 1 desde el usuario tienen la forma pet. XXXXX o resp. XXXXX. Las interacciones del servicio de la etapa 1 hacia el usuario tienen la forma ind. XXXXX o conf. XXXXX.

Se observan las siguientes directrices al elaborar estos diagramas de flujo:

- Las columnas verticales representan cada una de las entidades funcionales identificadas en el modelo funcional para el servicio. Los flujos de información se muestran en orden descendente en el que se producen en el procesamiento de una llamada. El orden de las acciones de entidad funcional mostradas entre flujos de información no es significativo.
- Un flujo de información se caracterizará en los cronogramas (diagramas con flechas) como asociado con los términos petición/indicación o respuesta/confirmación, lo que se refleja en la primitiva que es comunicada al sistema de señalización subyacente ilustrado en la figura 8. El nombre de primitiva es, en general, una derivación directa del nombre de flujo de información. Los términos se muestran en asociación con el flujo de información para mostrar la relación entre el SDL de la etapa 2 y el SDL del sistema de señalización subyacente.

Un número de referencia identifica unívocamente un punto determinado en la secuencia de flujo de información de la etapa 2 y aparece en el diagrama de flujo de información en ese punto. También sirve de puntero a una descripción (véase 2.4 más adelante) de las acciones requeridas en este punto de la secuencia. Puede también aparecer una breve descripción de las acciones de entidad funcional en la parte correspondiente de los diagramas de flujo de información. El esquema de numeración de referencia a utilizar se describe a continuación.

Cada número es de la forma XYZ, y es un número asignado por el redactor de la descripción de la etapa 2, que identifica un punto determinado de la descripción de procedimiento de la etapa 2 (cronogramas y SDL) en el que se describen las acciones de entidad funcional. En las acciones de servicios suplementarios, X del número FEA debe ser "9". Y es el número de la FE en el que se ejecuta la acción. Z enumera las acciones en una sola FE (Z=1, ..., 9, A, ..., Z, a, ..., z). Este número es único dentro de la descripción de la etapa 2 de un determinado servicio (todas las variantes).

NOTA 6 – Los flujos de llamada básica se muestran con líneas de trazo discontinuo y corchetes angulares. Los flujos de servicio suplementario se muestran con líneas de trazo continuo y cabezas de flecha. La información transferida "en banda" se muestra con líneas de doble trazo interrumpido.

NOTA 7 – Las relaciones entre FE pueden también mostrarse en el diagrama.

2.3.2 Definición de los distintos flujos de información

Se determina el significado semántico y el contenido de información de cada flujo de información. Se puede identificar si un flujo de información necesita una confirmación y, en tal caso, ello hace necesario el retorno de un flujo de información del mismo nombre.

Los flujos de información con confirmación toman la forma de una petición de acción (en un sentido) y de una confirmación de que la acción se ha llevado a cabo (en el sentido de vuelta). Los flujos de información con confirmación son necesarios típicamente con fines de sincronización. Los dos casos más destacados son los de petición de asignación y/o de liberación de un recurso compartido.

Cuando las entidades funcionales que interactúan se hallan en sitios físicos distintos, los flujos de información serán llevados a cabo normalmente por protocolos del sistema de señalización. Cuando las entidades funcionales que interactúan se hallan en el mismo sitio, los flujos de información son internos y no repercuten en los protocolos del sistema de señalización.

Los cuadros deben construirse de manera que muestren todos los conceptos en cada flujo de información. Debe indicarse si cada concepto es o no obligatorio u opcional, y con qué relación se cursa la información (véase el cuadro 2).

Cuadro 2/Q.65 – Definición de los distintos flujos de información

PET.INFO1

Relación	Concepto	IND.PET	RESP.CONF
r _a , r _b , r _c	Petición de número facturado	Obligatorio	–
r _a , r _b , r _c	Número facturado	–	Obligatorio
r _a , r _b , r _c	Petición de tipo de servicio	Opcional	–
r _a , r _b , r _c	Tipo de servicio	–	Opcional

2.4 Paso 4 – Acciones de las entidades funcionales (véase en 3.3 el paso 4 alternativo que utiliza técnicas orientadas a objetos)

Las acciones de la etapa 2 llevadas a cabo en una unidad funcional, desde la recepción de cada flujo de información hasta la transmisión del siguiente flujo de información resultante, se identifican y enumeran. Quedan incluidas todas las acciones que aparecen externamente (las que son notificadas explícita o implícitamente a otras entidades funcionales). Las acciones que se identifican se representan en los diagramas de flujo de información, así como en los diagramas SDL, por medio de frases sencillas, o aparte, por medio de números de referencia.

NOTA – La implementación de los principios de tarificación de las Recomendaciones de la serie D es de competencia nacional. Por tanto, las FEA que únicamente son de tarificación deben incluirse sólo si existe necesidad directa de un soporte de señalización, o si el servicio puede ser proporcionado por redes públicas a través de una frontera internacional. En los casos en que los principios de tarificación de las Recomendaciones de la serie D necesiten reflejarse en descripciones de servicios de la etapa 2, debe introducirse la siguiente nota de pie de página para cada FEA que contenga tarificación.

"Esta FEA describe acciones que podrán ser adoptadas por las Administraciones para implementar los principios de tarificación de las Recomendaciones de la serie D."

Esta subcláusula de acciones de las entidades funcionales contiene descripciones de las acciones requeridas para cada entidad funcional y cada FEA es identificada por un número de referencia:

La forma de presentación de las acciones de las entidades funcionales se ilustra en la figura 9.

FEA de FE2

921:

- Interactuar con usuario para acumular información
- Seleccionar recurso de acceso a la red
- Facilidades de reserva, ambos sentidos si es necesario

922:

- Interactuar con usuario para obtener dirección de llamada
- Determinar e indicar fin de marcación

Figura 9/Q.65 – Ejemplo de descripciones de acciones de entidades funcionales

2.5 Paso 5 (opcional) – Diagramas SDL para entidades funcionales

El lenguaje de descripción y especificación del sistema (SDL, véase UIT-T Z.100) en su representación gráfica se utiliza para describir formalmente un determinado servicio en términos de las acciones realizadas en las entidades funcionales que intervienen en la prestación del servicio y los flujos de información resultantes de estas acciones o que las desencadenan. Los diagramas SDL se basan en (y son coherentes con) los diagramas de flujo de información generados en el paso 2 del método de descripción de servicios, y describen con detalle la explotación normal, infructuosa y anormal del servicio.

Los flujos de información, que se definen completamente en los aspectos de su contenido y sus puntos de origen y de destino como parte del diagrama SDL, son soportados por procedimientos de señalización especificados en descripciones de servicios de la etapa 3.

La descripción del SDL figura en UIT-T Z.100. En la cláusula siguiente figura un resumen de sus elementos.

NOTA – El paso 5 (SDL) se considera opcional para la etapa 2 solamente si se especifican diagramas SDL formales en la descripción de la etapa 3.

2.5.1 Aspectos generales del SDL

Los elementos de una descripción de servicio en SDL son sistemas, bloques, procesos y procedimientos (véase 2.4/Z.100 y anexo B/Z.100). Los sistemas, bloques y procesos pueden definirse como una instanciación del tipo de sistema, tipo de bloque o tipo de proceso. Un (tipo de) sistema, (tipo de) bloque, (tipo de) proceso o procedimiento se describe en uno o más diagramas. Los diagramas de procedimiento aparecen si los diagramas de proceso incluyen llamadas de procedimiento.

La información entre bloques o entre procesos se transporta en señales que pueden tener parámetros que contengan datos. Los datos se definen en definiciones de tipo de datos.

2.5.1.1 El diagrama del sistema

En el contexto de la descripción de la etapa 2 de un determinado servicio de telecomunicación, el diagrama de tipo de sistema describe el sistema en forma de bloques, canales y señales. Los bloques representan el conjunto de funciones de red y de acceso de red que necesita ejecutar el sistema a fin de proporcionar un servicio. Las señales transportadas por canales (véase 2.5/Z.100) describen la información intercambiada entre bloques y entre bloques y el entorno del sistema. Los canales se conectan a bloques o a la trama envolvente de un diagrama a través de las denominadas puertas si el bloque se define como una instancia de un tipo de bloque. La lista de señales cursadas por los bloques del sistema forma parte del diagrama del sistema.

Un ejemplo de diagrama de un sistema denominado BasicService (que se supone que es el servicio portador con conmutación de circuitos) se muestra en las figuras 10a y 10b. El diagrama en este caso abarca dos tramas denominadas "definición" (figura 10a) y "estructura" (figura 10b).

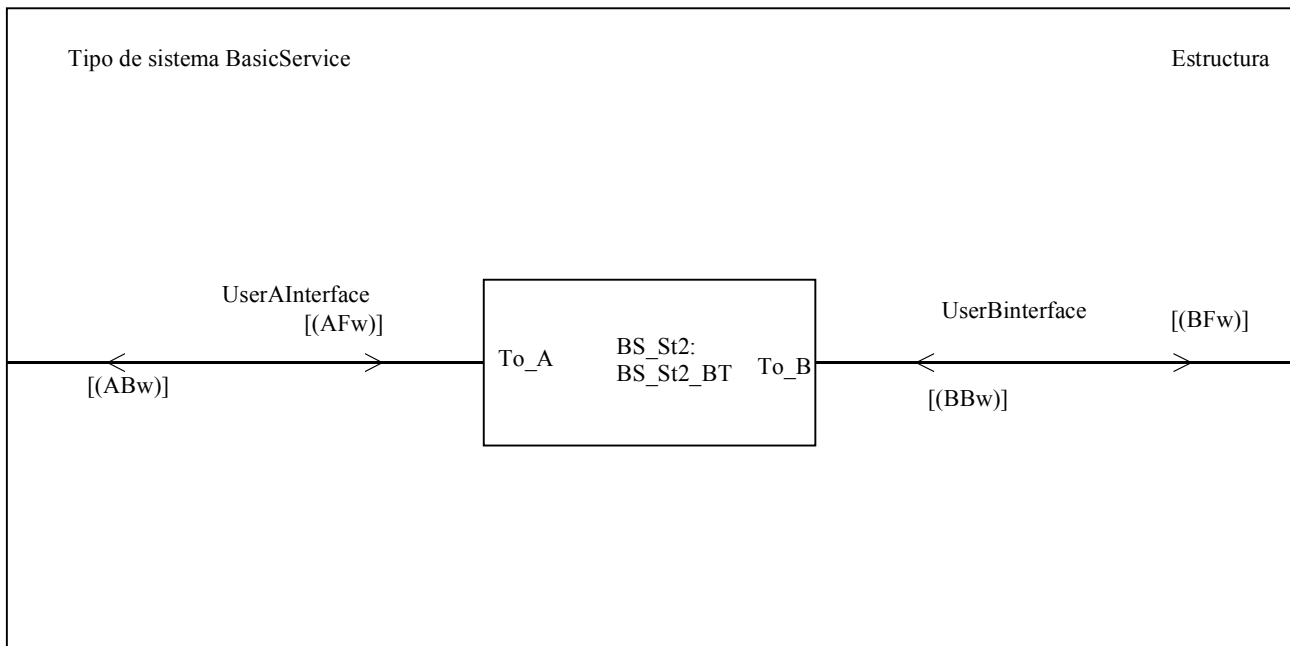
NOTA – El número de tramas utilizadas para dibujar un diagrama puede elegirse arbitrariamente y suele basarse en el espacio de composición disponible y en consideraciones de legibilidad.

La trama designada "definición" contiene:

- a) Un símbolo de tipo de bloque que contiene el número del tipo de bloque (BS_St2_BT) precedido por la palabra clave "virtual". Esto significa que el tipo de bloque BS_St2_BT puede ser redefinido, por ejemplo, cuando resulta necesario incorporar lógica de servicio adicional en el servicio básico para soportar un servicio suplementario.
- b) Un símbolo de texto que contiene la lista, encabezada por la palabra clave SIGNAL de las señales que pueden transferirse entre el sistema y el entorno.
- c) Símbolos de texto que contienen descripciones de listas de señales (palabra clave SIGNALLIST) denominadas Afw, Abw, Bfw y Bbw, respectivamente.
- d) Un símbolo de texto que contiene la definición (palabras clave NEWTYPE, ENDNEWTYPE) de un tipo de datos RejectCause. Las variables de este tipo pueden asumir los valores (palabra clave LITERALS) RequestNotValid, NoRoute, NoBresources, NoResponse y Busy. Los tipos de datos normalizados [por ejemplo, booleano, entero, (véase el anexo D/Z.100)] no necesitan definirse de nuevo en las descripciones de servicio.
- e) Un símbolo de texto que contiene una lista de sinónimos que no se especifican en la descripción de servicio. La palabra clave SYNONYM indica que se dan nombres a valores

definidos externamente (palabra clave EXTERNAL), por ejemplo, NoAnswerTime que es del tipo de datos normalizado Duration significa un valor que está definido externamente.

La trama designada "estructura" describe los bloques y canales del sistema BasicService. En este caso el sistema contiene una instancia BS_St2 del tipo de bloque BS_St2_BT que abarca todas las acciones de red relacionadas con el servicio básico. Los usuarios del servicio están situados fuera del sistema, es decir, en el entorno. Sus acciones no se describen en la etapa 2; sólo sus interacciones con la red. Éstas son representadas por las señales definidas en las listas de señales y son transferidas por canales denominados hacia y desde el entorno. Así, las señales de las listas de señales AFw y ABw son transferidas hacia y desde el entorno por el canal denominado UserAInterface, que está conectado a la instancia de bloque BS_St2 en la puerta To_A. Análogamente las señales de las listas de señales Bfw y BBw son transferidas desde y hacia el entorno por el canal denominado UserBInterface, que está conectado a la instancia de bloque BS_St2 en la puerta To_B.



T1182260-96

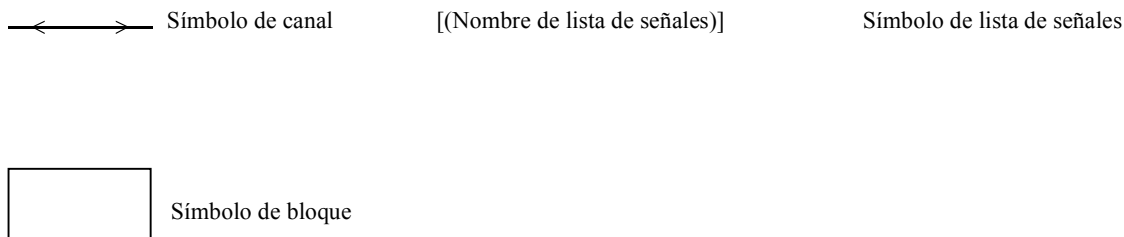


Figura 10b/Q.65 – Ejemplo de diagrama de estructura de tipo de sistema

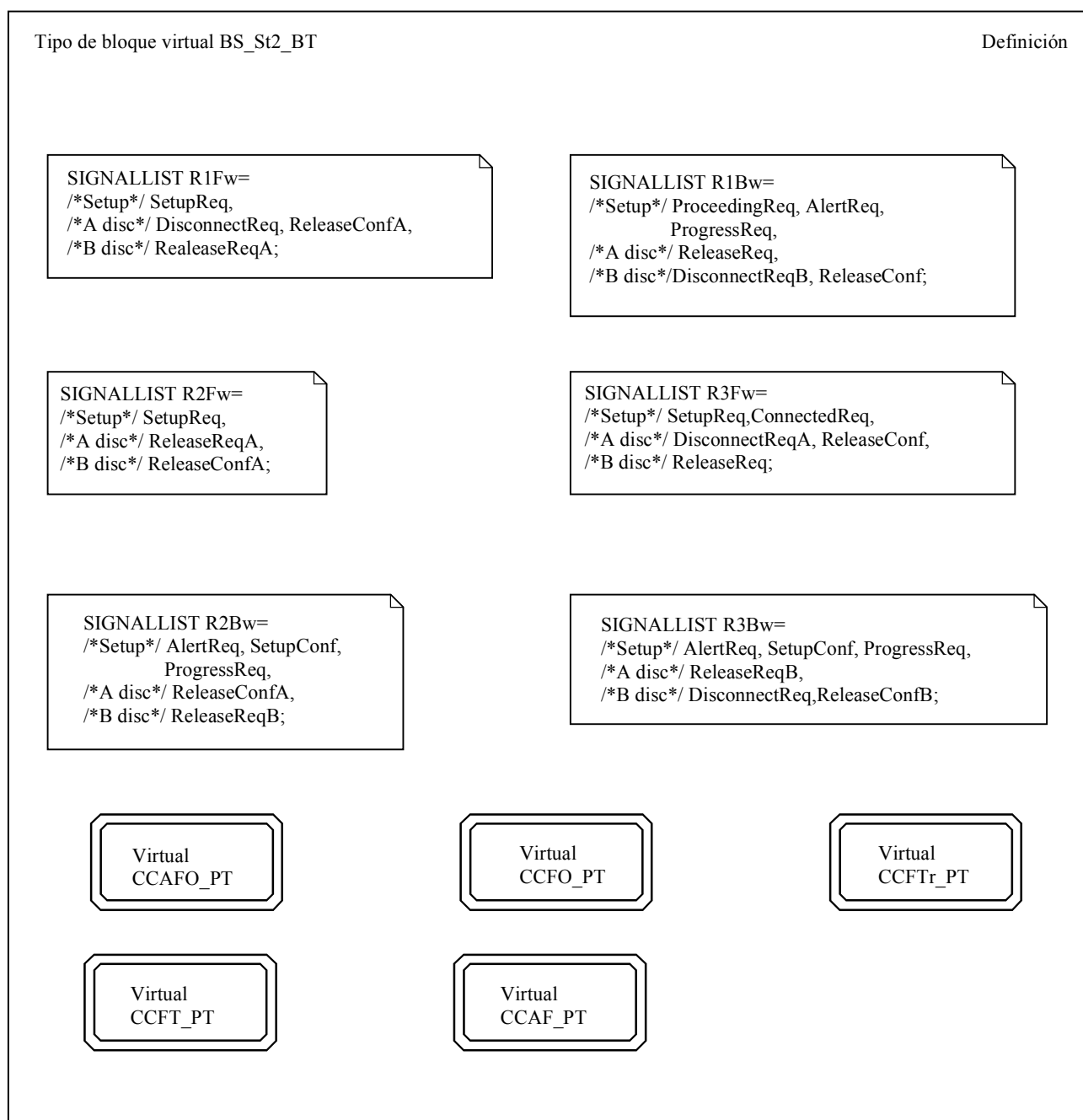
2.5.1.2 El diagrama de bloques

En el contexto de una descripción de servicio de la etapa 2, el diagrama de tipo de bloques define los tipos de proceso utilizados en el sistema y la forma en que las instancias de estos tipos se estructuran y se comunican a fin de soportar un servicio. Las tramas de las figuras 11a (definición) y 11b (estructura) representan el diagrama para el tipo de bloque BS_St2. Como ya se advirtió en el diagrama de tipo de sistema, este tipo de bloque ha sido definido como el tipo virtual para permitir redefinición si es necesario. El bloque abarca instancias de 5 tipos de proceso, ubicadas en las 5 entidades funcionales (es decir, las funciones de agente de control de llamadas de origen y terminación, y las funciones de origen, tránsito y terminación de control de llamada) que intervienen en la iniciación y terminación de una instancia del servicio básico o, en otras palabras, en el establecimiento y liberación de una llamada básica.

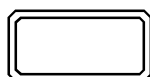
En el ejemplo del servicio básico, los 5 tipos de procesos se han definido como tipos virtuales, para permitir redefinición futura (por ejemplo, para insertar lógica de servicio suplementario).

En el diagrama de proceso, las transferencias de información se producen por rutas de señales. Éstas conectan instancias de proceso a canales a través de puertas denominadas. Por ejemplo, la ruta R1 interconecta instancias de proceso CCAFO y CCFO vía puerta o OG en CCAFO y puerta IC en CCFO y la ruta UIA conecta instancias de proceso CCAFO al canal UserAInterface vía puerta To_A. Las señales transferidas a lo largo de cada ruta se describen en las listas de señales.

Los símbolos de proceso contienen los nombres de la instancia y tipo de proceso separados por un signo dos puntos y dos números, por ejemplo (1,1), siguiendo al nombre de instancia de proceso. Éstos representan el número de instancias del proceso que existen cuando se crea el sistema y el número máximo de instancias simultáneas del proceso, respectivamente.



T1182270-96



Símbolo de tipo de proceso

Figura 11a/Q.65 – Ejemplo de diagrama de definición de tipo de bloque

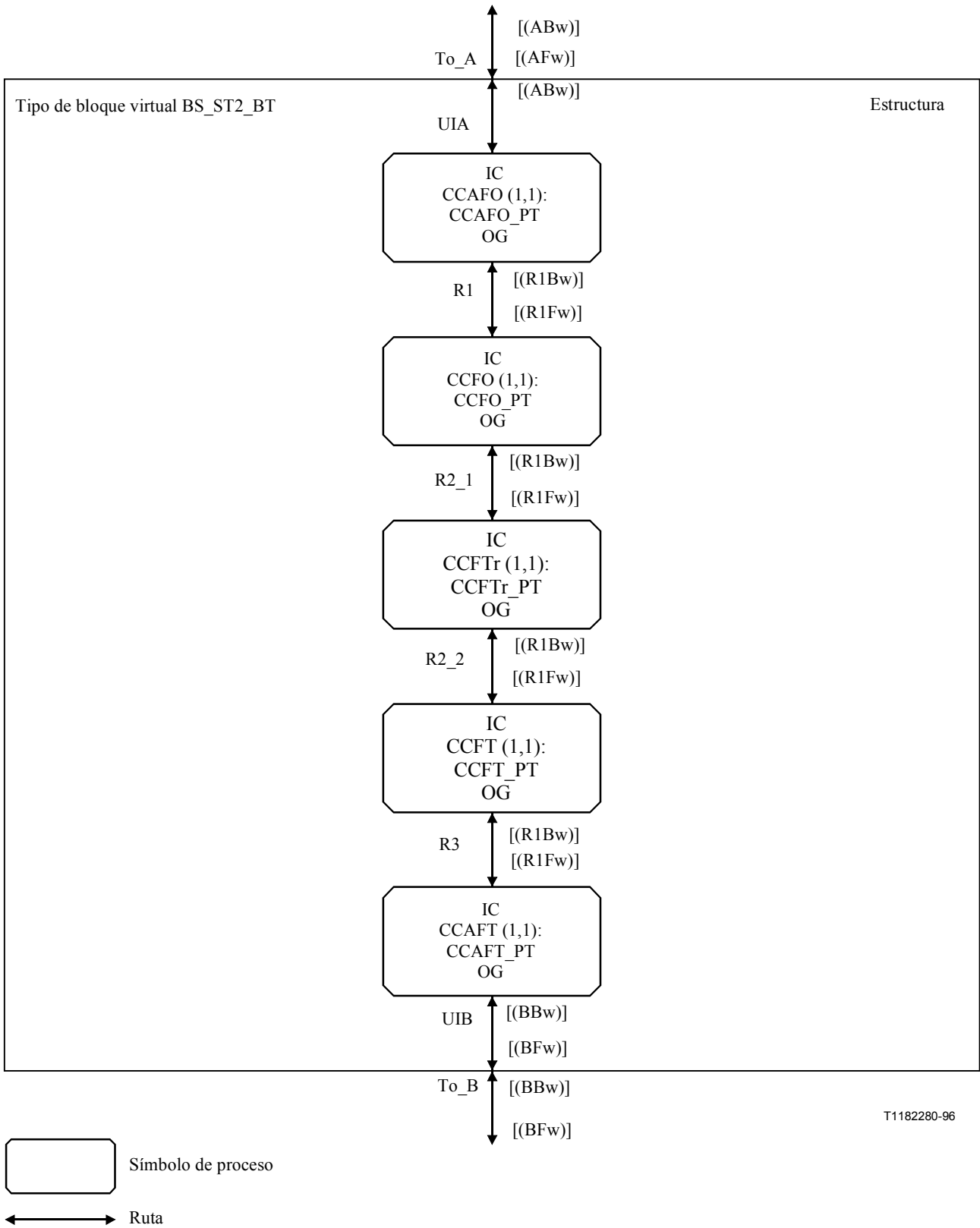







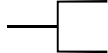

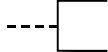

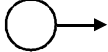








Figura 11b/Q.65 – Ejemplo de diagrama de estructura de tipo de bloque

2.5.1.3 El diagrama de proceso

Un diagrama de proceso describe la secuencia de acciones (transacciones) que son ejecutadas por una instancia de proceso cuando se recibe una señal. El cuadro 3 muestra los símbolos gráficos que se utilizan más frecuentemente en los diagramas de proceso de las descripciones de servicio. Los números entre corchetes indican la cláusula de UIT-T Z.100 en la que se describen estos símbolos más detalladamente.

Cuadro 3/Q.65 – Símbolos SDL

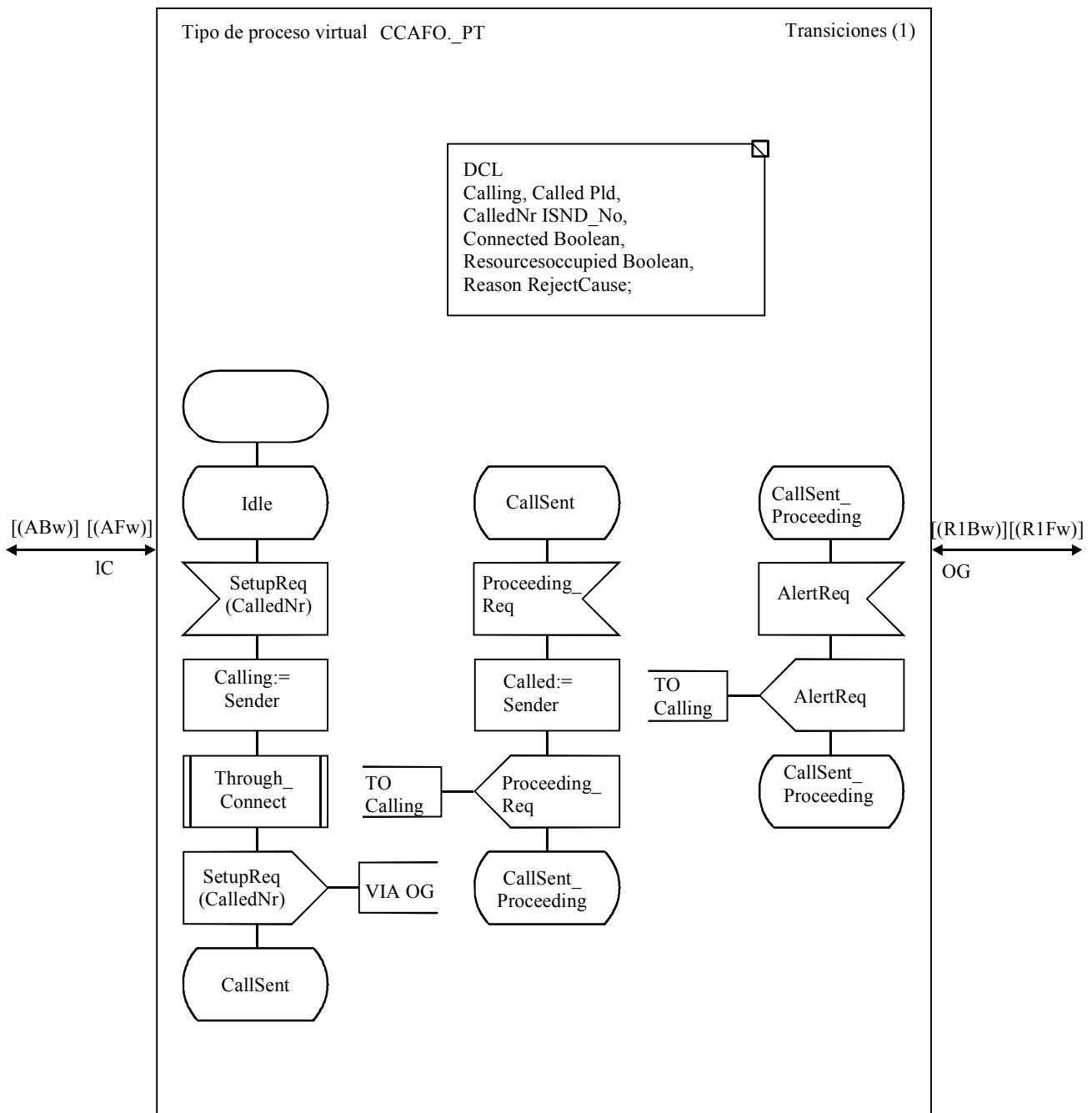
	Arranque [2.6.2]		Opción de transición [4.3.4]
	Estado [2.6.3]		Conservación [2.6.5]
	Entrada [2.6.4]		Condición habilitante [4.12]
	Salida [2.7.4]		Ampliación de texto [2.2.7]
	Tarea [2.7.1]		Comentario [2.2.6]
	Llamada a procedimiento [2.7.3]		Conector-entrada [2.6.7]
	Procedimiento [2.4.6]		Conector-salida [2.6.8.2.2]
	Arranque procedimiento [2.4.6]		Retorno (de un procedimiento) [2.6.8.2.4]
	Decisión [2.7.5]		Parada (término de un proceso) [2.6.8.2.3]

T1182290-96

Se muestran dos ejemplos de diagramas de proceso. La figura 12a contiene un fragmento de tipo de proceso virtual CCAFO_PT de tipo de bloque BS_St2_BT y la figura 12b un fragmento de tipo de proceso virtual CCFT_PT (véanse las figuras 11a y 11b).

En relación con la figura 12, se aplican las siguientes notas aclaratorias:

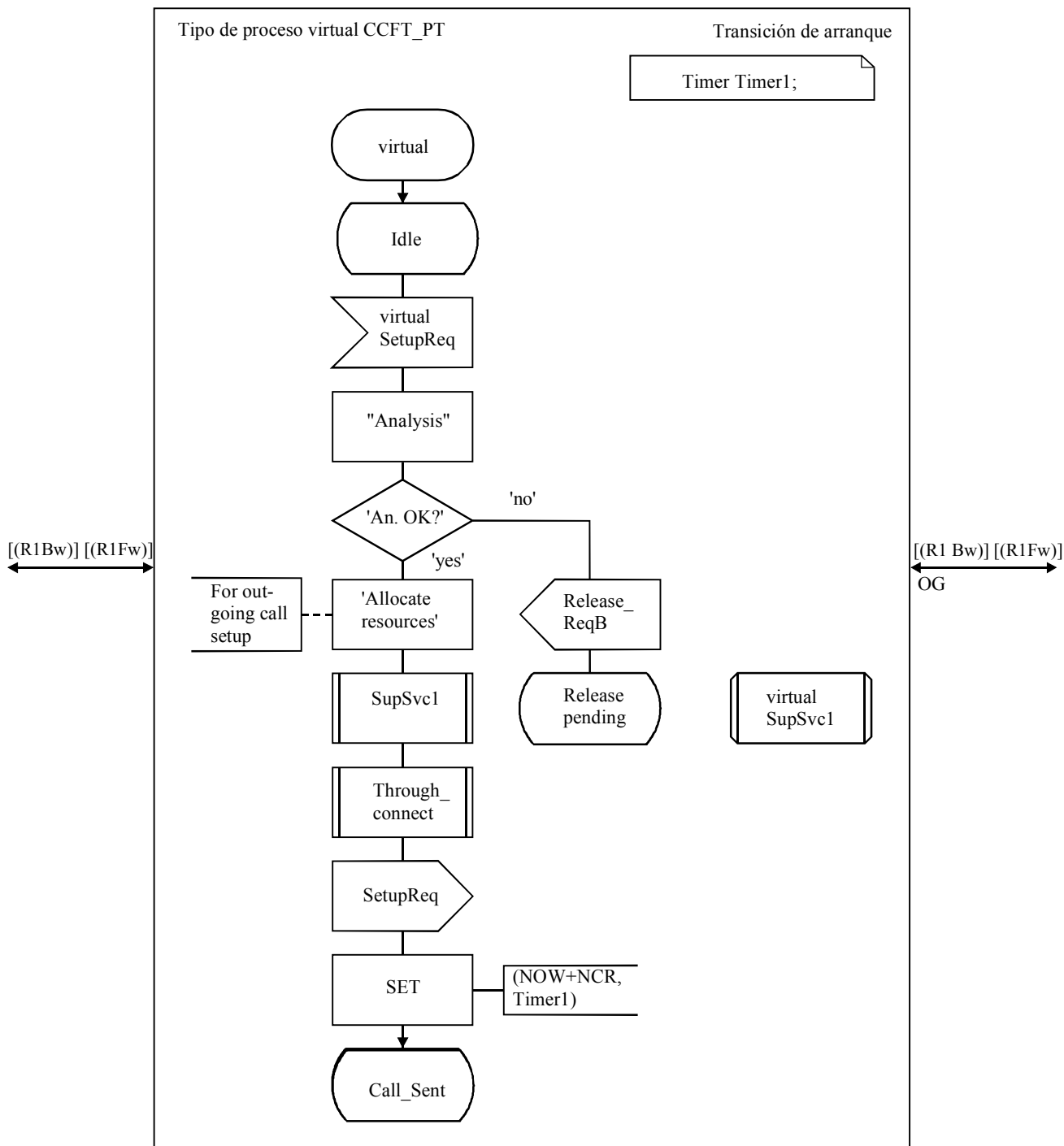
- El símbolo de texto contiene la declaración (palabra clave DCL) de las variables utilizadas por el proceso. Para cada variable se define un nombre y tipo de datos, por ejemplo, las variables denominadas llamante y llamada son del tipo predefinido Pid (identificador de instancia de proceso) y se utilizan por tanto para identificar un proceso. También de un tipo predefinido (es decir, booleano) son las variables Connected y Resourcesoccupied, mientras que las variables CalledNr y Reason son de los nuevos tipos ISDN_No y RejectCause, respectivamente, que son específicos de esta descripción de sistema.
- Los símbolos de entrada y salida contienen el nombre de la señal SDL que representan. La expresión (o expresiones) que sigue al nombre de señal, por ejemplo, SetupReq (CalledNr) indica información (un parámetro) transportada por la señal.
- Sender, como por ejemplo, en el símbolo de tarea Calling:=Sender, es una palabra clave utilizada para identificar el proceso de envío de la señal que activó la transición en curso. En el ejemplo de los dos procesos de la figura 13, Calling y Called, se identifican de este modo. En este caso particular, estos procesos están en el entorno y no se describen en el ejemplo de descripción de sistema.



T1182300-96

Figura 12a/Q.65 – Ejemplo 1 de proceso de sistema – Fragmento del proceso CCAFO

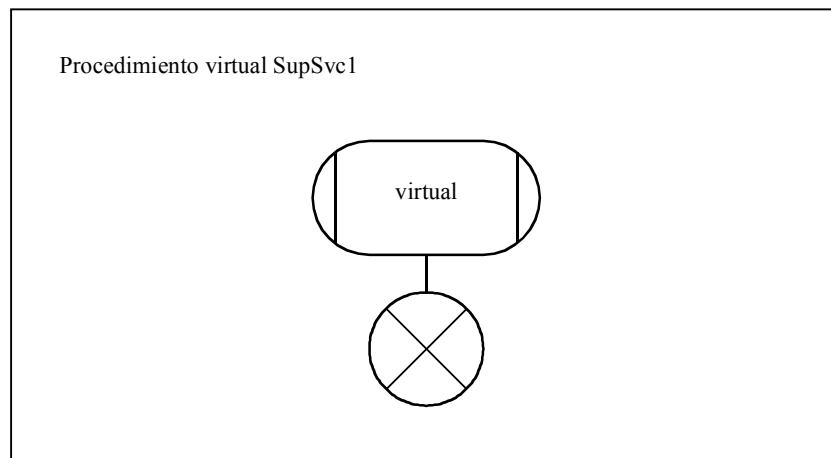
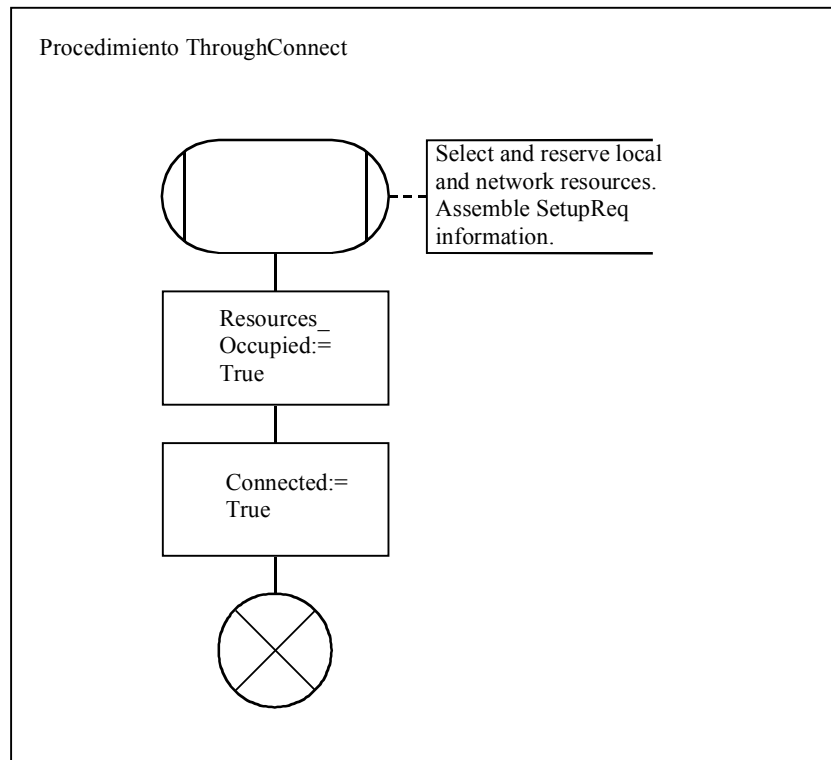
La figura 12b ilustra el método de crear una apertura para mejoras futuras en una transición existente. La apertura se obtiene insertando una llamada a procedimiento, por ejemplo, al procedimiento SupSvc1 en el ejemplo mostrado, en un punto de la transición en el que se prevé la adición posterior de lógica. Inicialmente el procedimiento llamado es "vacío". Se redefine para proporcionar lógica adicional siempre que resulte necesario (véase 2.5.2), por cuya razón se define como virtual.



T1182310-96

**Figura 12b/Q.65 – Ejemplo 2 de un diagrama de proceso –
Fragmento de tipo de proceso CCFT**

2.5.1.4 El diagrama de procedimiento



T1182320-96

Figura 13/Q.65 – Ejemplos de diagrama de procedimiento

Si los procedimientos son llamados por tipos de proceso del sistema, estos procedimientos se describen entonces en diagramas de procedimiento. Los ejemplos de la figura 13 muestran el procedimiento ThroughConnect llamado por las instancias de procesos CCAFO y CCFT y el procedimiento SupSvc1 llamado por la instancia de proceso CCFT. Como se prevé que este último sufrirá modificación, se ha definido como un procedimiento virtual. Como se ha señalado antes, el procedimiento SupSvc1 es vacío y actúa por el momento simplemente como ocupante del lugar para la adición de lógica de servicio en el futuro.

Los símbolos utilizados en los diagramas se explican en el cuadro 3.

2.5.2 Adición de funcionalidad a las descripciones de servicios de la etapa 2 existentes

2.5.2.1 Especialización

Surge ocasionalmente la necesidad de adición de una nueva funcionalidad a una descripción de servicios existente. Un caso típico es el conjunto de modificaciones requeridas para la descripción del servicio portador básico con conmutación de circuitos (servicio básico) cuando se normaliza un nuevo servicio suplementario.

Cuando la descripción de un nuevo servicio (por ejemplo un servicio suplementario) requiere las modificaciones de la descripción de un servicio existente (por ejemplo, el servicio básico), puede utilizarse el concepto de una especialización. Una especialización permite describir los procedimientos lógicos asociados con un nuevo servicio sin afectar a la descripción del servicio existente mediante provisión, por ejemplo, de la capacidad de:

- añadir y/o redefinir tipos de bloques en diagramas de sistemas;
- añadir canales en diagramas de tipos de sistemas para transportar señales existentes y/o nuevas;
- añadir y/o redefinir tipos de procesos en diagramas de bloques;
- añadir rutas en diagramas de tipos de bloque para transportar señales existentes y/o nuevas;
- añadir nuevos estados y transiciones de estados en los diagramas de tipos de proceso;
- añadir nuevas transiciones desencadenadas por nuevas señales a estados existentes;
- redefinir transiciones de estados existentes; o
- redefinir procedimientos.

En SDL, la redefinición se habilita definiendo como virtuales aquellos tipos de bloque, tipos de proceso, procedimientos y transiciones que probablemente sufran cambio a medida que evoluciona el sistema. En las figuras 10a, 11a, 12b y 13 se muestran ejemplos de definiciones virtuales.

2.5.2.2 Descripción de servicios suplementarios

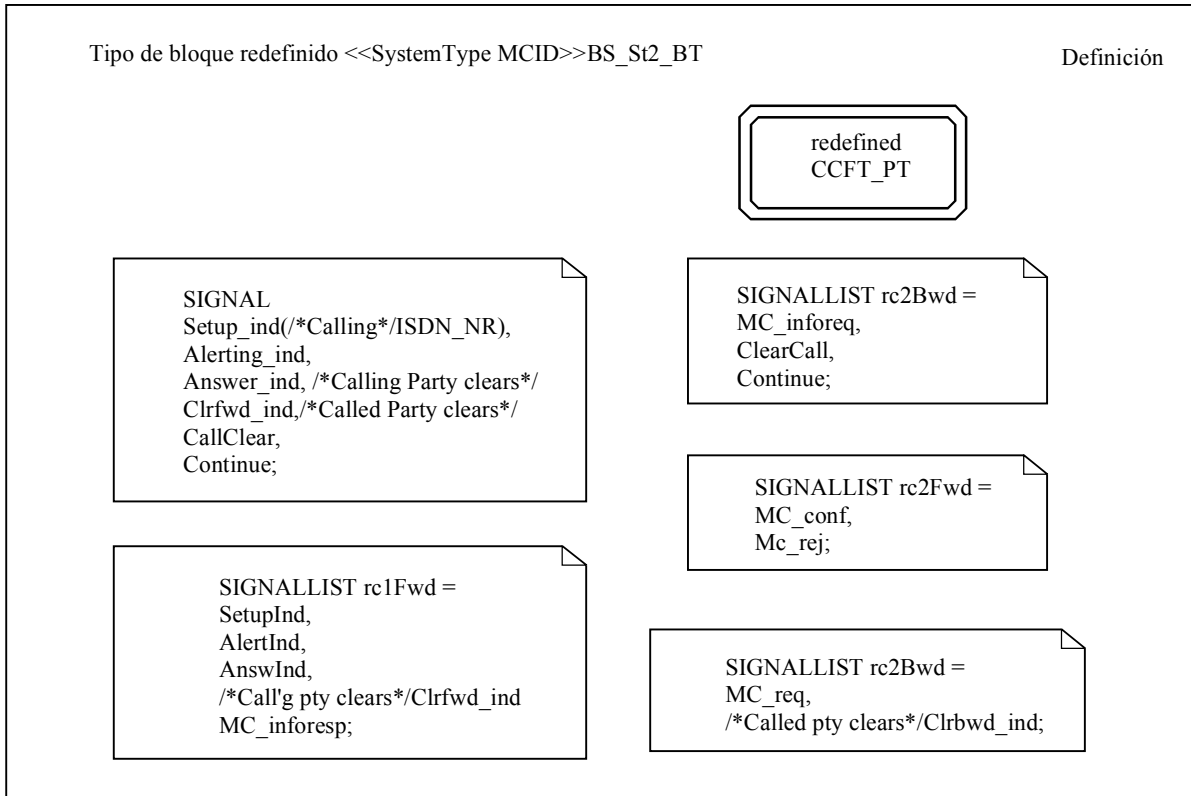
Un servicio suplementario se describe modificando o redefiniendo la lógica de servicios básicos. Por esta razón los tipos de bloques de servicio básico, los tipos de procesos y las transiciones que probablemente sean afectados por la adición de un servicio suplementario se designan por "virtuales" (véase 2.5.2.1).

Los cambios al servicio básico causados por la introducción de un servicio suplementario se describen entonces declarando que el "servicio suplementario" hereda el tipo de sistema "servicio básico" y por adición de bloques, procesos, transiciones, señales, etc., nuevos o redefinidos requeridos para soportar el servicio suplementario. "Hereda" es una palabra clave de acuerdo con UIT-T Z.100.

El método de descripción se ilustra en las figuras 14 a 17, utilizando como ejemplo un servicio de identificación de llamadas maliciosas simplificado.

El diagrama de sistema de la figura 14 indica que:

- a) el tipo de sistema MCID se basa en (hereda) el sistema BasicService;
- b) se ha redefinido el bloque BS_St2 del sistema BasicService;
- c) se ha añadido el canal MCIDUserB sobre el cual el bloque BS_Stage2 intercambia señales MC_req, Mc_conf y MC_rej con el entorno (usuario B en este caso);
- d) se ha introducido un nuevo tipo de datos MC_rejcause que puede asumir los valores indicados (literales).



T1182340-96

**Figura 15a/Q.65 – Tipo de bloque BS_St2 (definición)
redefinido para el sistema MCID**

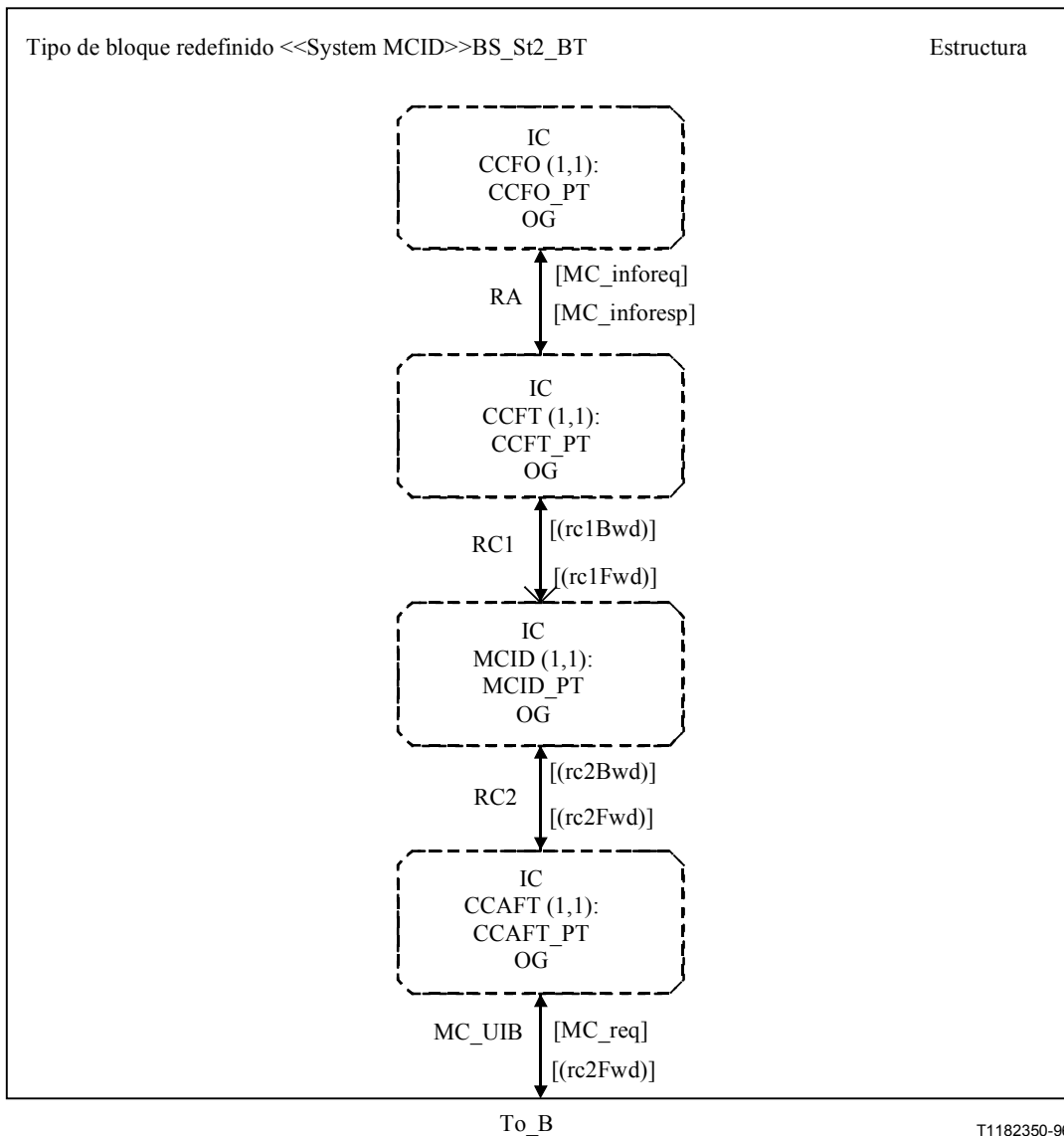
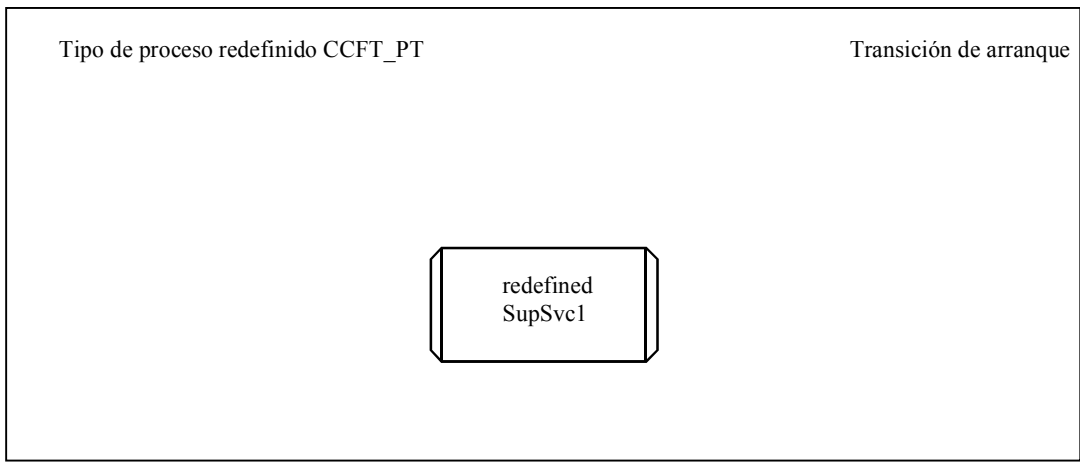


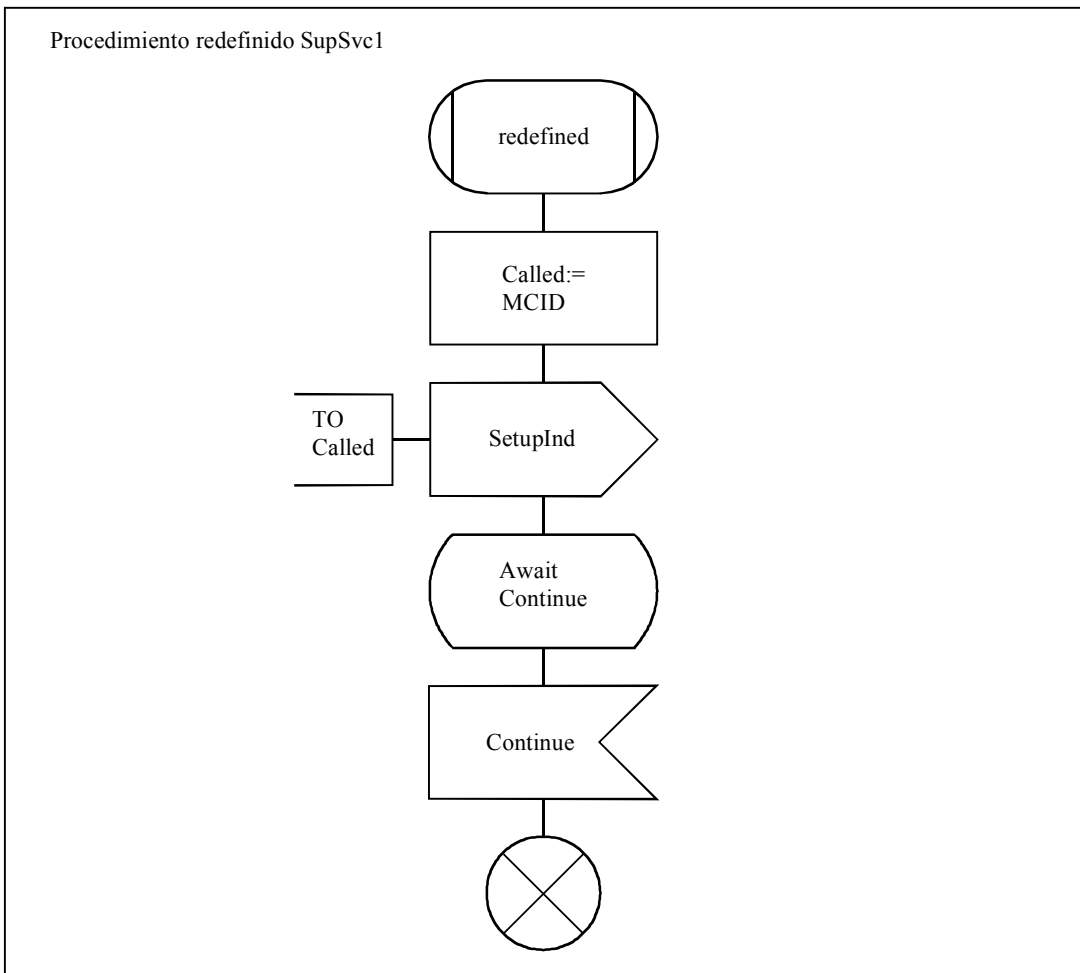
Figura 15b/Q.65 – Tipo de bloque BS_St2 (estructura) redefinido para el sistema MCID

La figura 16 muestra la redefinición del tipo de proceso CCFT_PT. El único cambio que necesita mostrarse aquí es la indicación de que el procedimiento SupSvc1 que se define como virtual en el tipo de sistema BasicService está siendo redefinido. El procedimiento redefinido se muestra en la figura 17.



T1182360-96

Figura 16/Q.65 – Tipo de proceso CCFT_PT redefinido para el sistema MCID



T1182370-96

Figura 17/Q.65 – Procedimiento SupSvc1 redefinido para el sistema MCID

2.5.3 Lotes

SDL 92 ofrece la posibilidad de utilizar tipos definidos en un sistema también en otros sistemas. Esto se consigue proporcionando un diagrama de lote que sirve de contenedor de conceptos tales como definiciones de tipos de sistema, definiciones de tipos de bloques, definiciones de tipos de proceso, señales o sinónimos.

La utilización de un tipo que ha sido definido en un lote se indica poniendo, en un símbolo de texto por encima de un diagrama, el número de lote precedido por la palabra utilización. La figura 18 contiene un ejemplo de una definición de lote y de la utilización de un tipo definido en el lote. La figura refleja el hecho de que el tipo de bloque B1 en el BCSistema se define utilizando el tipo de bloque Block1_BT en el lote BCBlocks.

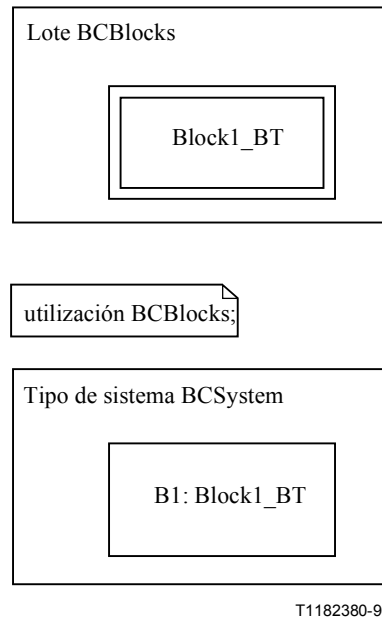


Figura 18/Q.65 – Ejemplo de la definición de un tipo de bloque en un lote y de la utilización posterior del lote en un sistema

2.6 Paso 6 – Asignación de las entidades funcionales a emplazamientos físicos (escenarios)

En el paso 1 se define para cada servicio básico y suplementario un modelo funcional formado por entidades funcionales, cada una de las cuales tiene una relación bien definida con las demás. El paso 6 consiste en asignar dichas entidades funcionales a sitios físicos y en definir todas las realizaciones físicas de interés a las que en lo sucesivo se denomina escenarios.

Para un modelo funcional se podrá definir más de un escenario, de forma que las Administraciones puedan elegir dónde se ofrecerá el servicio. Por ejemplo, una entidad funcional de servicio suplementario podrá estar situada en una CAP o en una central.

Para la asignación de las entidades funcionales habrá que tener en cuenta que:

- en principio, una entidad funcional puede asignarse a cualquier emplazamiento físico;
- diversas unidades funcionales pueden asignarse a un mismo emplazamiento físico;
- se deben definir, para cada servicio suplementario, escenarios de red que incluyan el emplazamiento de sus entidades funcionales del servicio básico;
- los diferentes emplazamientos físicos de las entidades funcionales pueden dar lugar a pequeñas diferencias en las capacidades de los nodos (por ejemplo, las acciones de

conmutación del trayecto de transmisión pueden depender de que el acceso se haga en una central o en una PBX);

- e) las relaciones entre pares de entidades funcionales, de acuerdo con el modelo funcional que se utilice, deberán ser invariables para todos los escenarios recomendados.

El apartado e) implica, por ejemplo, que los flujos de información de un servicio suplementario no serán afectados por la reasignación de una o más entidades funcionales requeridas de una central de la red pública a una PBX, o viceversa.

Todos los escenarios identificados se considerarán en la etapa 3 para la definición de los protocolos de señalización, las capacidades de conmutación y las capacidades de servicio.

Posibles emplazamientos físicos y su correspondiente representación simbólica son:

- equipo terminal: tipo 1 o adaptador de terminal TE;
- terminación de red; tipo 2: NTE (normalmente en una PBX);
- central local: LE;
- central de tránsito: TR;
- punto de conmutación del servicio: SSP;
- punto de control del servicio: SCP;
- base de datos: DB;
- periférico inteligente: IP.

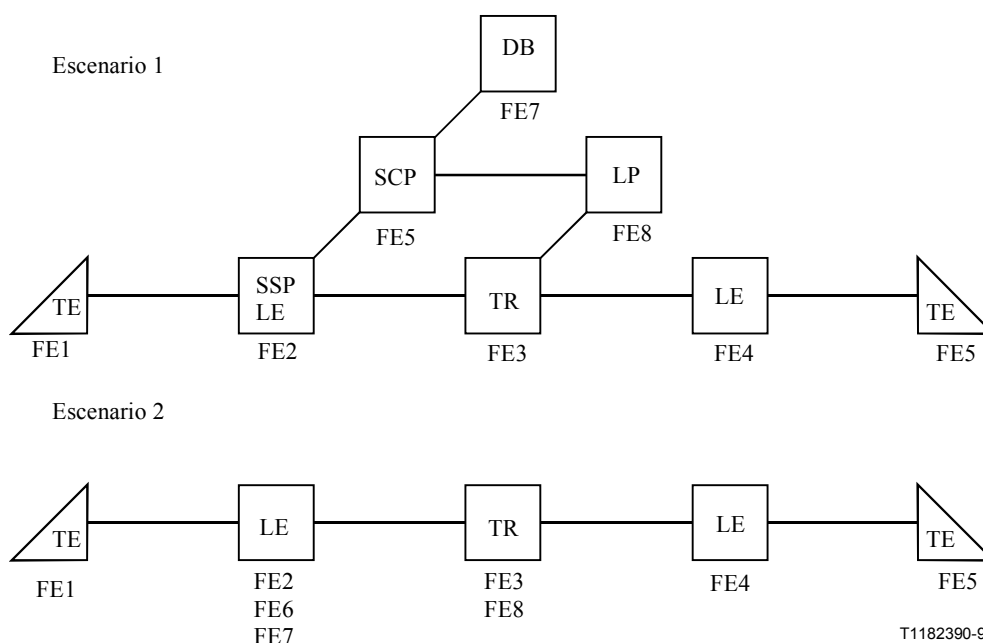


Figura 19/Q.65 – Ejemplo de dos escenarios para asignar entidades funcionales a emplazamientos físicos

3 Pasos alternativos que utilizan técnicas orientadas a objetos

Las etapas o pasos a seguir para desarrollar servicios/aplicaciones desde el concepto hasta la realización física (protocolo) se han descrito en forma detallada en la cláusula 2. El propósito de esta subcláusula es describir métodos alternativos que utilizan técnicas orientadas a objetos (O-O). Es posible que se utilicen tanto técnicas O-O como técnicas que no lo son para describir las interfaces

de RI, sea utilizando métodos O-O en alguna interfaz y métodos que no son O-O en otra. El resultado sería una colección de llamadas a API (usando técnicas O-O) y una colección de flujos de información (usando técnicas que no son O-O). Esto resultaría ser bastante desordenado, aunque para ambos resultados se podría hallar su correspondencia a, por ejemplo, el protocolo de aplicación de red inteligente (INAP, *intelligent network application protocol*). De utilizarse ambos métodos, sería mejor usar uno de ellos para un conjunto de interfaces y el otro método para las interfaces restantes. La industria de la computación viene usando las técnicas orientadas a objetos desde hace algún tiempo, y corresponde por lo tanto que demos ejemplos de cómo se pueden utilizar dichos métodos para mejorar el desarrollo de las RI y facilitar su integración a entornos de computación distribuida. Hay muchos paquetes distintos de O-O que se pueden utilizar, siendo algunos mejores que otros. En esta cláusula se describirá el lenguaje de modelado unificado (UML, *unified modelling language*) y se lo adaptará a la descripción de alternativas a los pasos 2 a 4 de la metodología funcional unificada. El paso 1 sigue siendo el mismo.

3.1 Paso alternativo 2

Aplicaciones a paquetes (opcional) (véase la figura 1)

En el UML el paquete es un mecanismo de aplicación general destinado a organizar los elementos de modelado en grupos. En nuestra metodología se construye la aplicación o el servicio a partir de cierto número de paquetes. Los paquetes se usan para disponer los elementos de modelado en trozos más grandes que pueden ser manipulados como un grupo. Los paquetes bien diseñados agrupan elementos semánticamente equivalentes, y que tienden a cambiar como un grupo.

Cada paquete debe tener un nombre que lo distingue de otros paquetes. Un *nombre* es una cadena de texto. Ese nombre, por sí mismo, se conoce como *nombre simple*; un *nombre de trayectoria* es el nombre del paquete en cuestión con un prefijo consistente en el nombre del paquete dentro del cual vive, si lo hubiera. Típicamente, se dibuja un paquete mostrando sólo su nombre tal como se indica en la figura 20. Al igual que con las clases, se pueden suplementar los paquetes con valores rotulados o con compartimentos adicionales para revelar sus detalles.

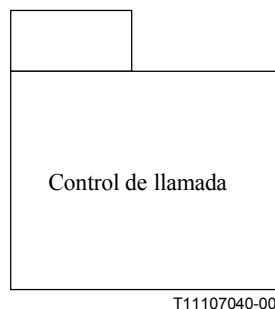


Figura 20/Q.65 – Ejemplo de paquete

En la realidad, la división en paquetes de una aplicación o de un servicio puede variar. Algunas aplicaciones constarán de numerosos paquetes, mientras que otras pueden consistir en un único paquete. Por ejemplo, la aplicación X de la figura 21 puede contener paquetes de control de llamada, autenticación y mensajería, mientras que una única aplicación de seguridad puede contener sólo un paquete de autenticación. Los paquetes pueden ser influidos por distintas aplicaciones; por ejemplo, la aplicación X puede requerir los subelementos a, b y c de autenticación, mientras que la aplicación Y requiere los subelementos b y d; por lo tanto, el paquete de autenticación resultante consistirá en los subelementos a, b, c y d (véase la figura 21).

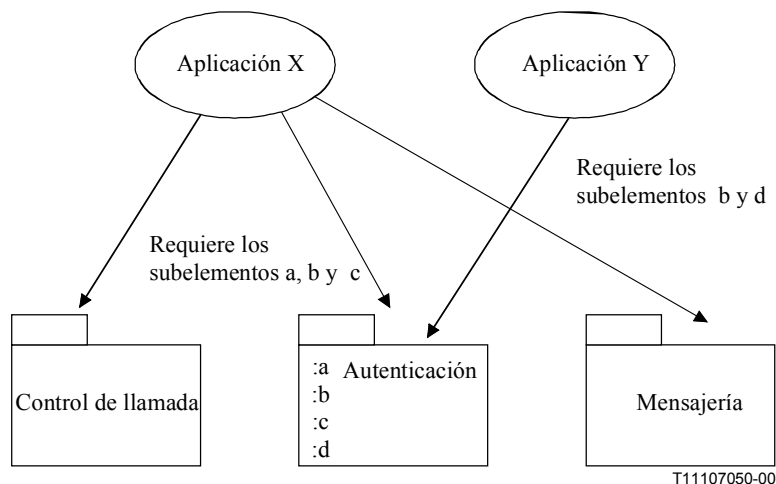


Figura 21/Q.65 – Correspondencia entre paquetes y aplicaciones

El método de los paquetes es una manera de describir las áreas que deben ser exploradas a fin de producir las clases interfaz necesarias, y agrupar varias clases interfaz que se necesitarán para una determinada interfaz a fin de ejecutar esa particular aplicación o servicio (véase el apéndice III.3.1, para obtener más detalles sobre este punto).

NOTA – Es posible omitir la etapa del "paquete" si se conocen las clases interfaz requeridas para implementar la aplicación/servicio.

3.2 Paso alternativo 3

Clases interfaz y diagramas de clase

3.2.1 Clases interfaz

Una interfaz es una colección -con nombre- de operaciones que se utiliza para especificar un servicio de una clase o de un componente. A diferencia de las clases o los tipos, las interfaces no especifican estructura alguna (por lo cual no pueden incluir ningún atributo), ni especifican una implementación (por lo cual no pueden incluir método alguno, que son los que proveen la implementación de una operación). Al igual que una clase, una interfaz puede tener un número cualquiera de operaciones. Estas operaciones se pueden suplementar con propiedades de visibilidad, propiedades de concurrencia, estereotipos, valores marcados y restricciones.

Se pueden dibujar las operaciones mostrando sólo su nombre, o se las puede ampliar hasta mostrar su signatura completa y otras propiedades como en la figura 22.

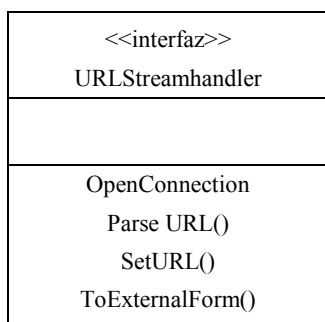


Figura 22/Q.65 – Operaciones en una clase interfaz

No se describe una clase objeto ya que ello daría los detalles de cómo implementar esta clase, es decir, especificaría cómo el sistema lleva a cabo el método que se ha invocado sobre él. Las clases Interfaz no describen métodos específicos a la implementación. Por lo tanto, las clases Interfaz se pueden implementar en distintos tipos de interfaces físicas y se pueden importar de otros sistemas para una interfaz determinada, sin por ello comprometer el procedimiento usado para llevar a cabo el método invocado.

Dependiendo de la funcionalidad, las clases interfaz se pueden describir en pares, la "clase interfaz" y la "clase interfaz de aplicación". La "clase interfaz básica" residiría en el sistema en un lado de la interfaz y la "aplicación" en el otro. Por lo general, los métodos invocados por la "aplicación" requerirían un resultado que es implementado por la "clase interfaz básica".

Se utilizan dos tipos útiles de clases interfaz para describir un sistema, la clase interfaz "marco" y la clase interfaz "servicio genérico".

3.2.1.1 Clases interfaz marco

Las clases interfaz marco son aquellas que describen una plantilla extensible para aplicaciones dentro de un dominio. Un marco es una especie de microarquitectura que abarca un conjunto de mecanismos que trabajan juntos para resolver un problema común para un dominio común.

Por lo tanto, viene bien pensar en una "clase interfaz marco" como una clase que se usa casi siempre en una interfaz, independientemente de la "clase servicio" que también se esté invocando. Un ejemplo del caso sería la autenticación. Es casi seguro que en cualquier comunicación a través de una interfaz, ambos extremos deberían inicialmente autenticar uno al otro. Otros ejemplos son "notificación de evento" y "gestión de integridad".

Las clases interfaz marco serían casi siempre descritas de a pares, por ejemplo, "Iauthentication" e "IAppAuthentication", según se describe en cláusula 3.2.1, donde la "I" con que comienza el nombre indica que se trata de una "clase interfaz" y no de una "clase objeto".

3.2.1.2 Clases interfaz servicio genérico

Las clases interfaz servicio genérico son aquellas clases que satisfacen los requisitos de servicio de la interfaz en cuestión. Por ejemplo, una clase servicio genérico típica sería "gestor de control de llamada". Los pares resultantes de clases interfaz serían "ICallControlManager" e "IappCallControlManager". Esta clase servicio genérico podría ser considerada como un "paquete".

Los dos distintos tipos de clase interfaz deberían ser definidos en forma detallada, es decir, se deben poblar con sus respectivas invocaciones de métodos. Pero antes de llegar a esta etapa final, se deben describir las relaciones entre las clases interfaz; esto se hace utilizando "diagramas de clase".

3.2.2 Diagramas de clase

Las relaciones entre las clases interfaz se muestran utilizando "diagramas de clase" [véase la figura 23].

Un diagrama de clase es un diagrama que muestra un conjunto de clases, interfaces y colaboraciones, y sus relaciones. Se pueden utilizar diagramas de clase para crear un modelo conforme a la vista de diseño estático del sistema. Esta vista soporta principalmente los requisitos funcionales de un sistema –los servicios que el sistema debería proveer a sus usuarios finales.

Los diagramas de clase contienen generalmente los siguientes elementos:

- *Clases*: una clase es una descripción de un conjunto de objetos que comparten iguales atributos, operaciones, relaciones y semántica. Una clase implementa una o más interfaces.
- *Interfaces*: una interfaz es una colección de operaciones que se usan para especificar un servicio de una clase o un componente.

- *Colaboraciones*: una colaboración es una sociedad de clases, interfaces y otros elementos que trabajan juntos para proveer un cierto comportamiento cooperativo superior a la suma de todas sus partes.
- *Relaciones de dependencia, generalización y asociación*: una "dependencia" consiste en utilizar una relación que determina que una modificación de las especificaciones de un elemento puede afectar a otro elemento que la utiliza, pero no necesariamente a la inversa. Una "generalización" es una relación entre un elemento general (llamado superclase o progenitor) y una especie más específica de ese elemento (llamado subclase o vástago). Por ejemplo, podríamos encontrarnos con la clase general "control de llamada" y con una especie más específica "control de llamada INAP". Una "asociación" es una relación estructural que describe un conjunto de enlaces, siendo un enlace una conexión entre objetos.

Los diagramas de clase son los diagramas más corrientes que se encuentran al crear modelos de sistemas orientados a objetos. Son importantes para visualizar los modelos estructurales, enunciar sus especificaciones y documentarlos, y también para construir sistemas ejecutables sea aplicando técnicas de diseño clásicas como reconstruyendo un comportamiento dado.

El diagrama de clases describe los tipos de objetos (implementaciones específicas) o clases interfaz del sistema, y los distintos tipos de relaciones estáticas existentes entre los mismos. Hay dos tipos principales de relaciones estáticas:

- asociaciones;
- subtipos.

Los diagramas de clase muestran también los atributos y operaciones de una clase, y las limitaciones que se aplican a cómo se conectan entre sí los objetos. En esta Recomendación no analizaremos los atributos, pues éstos son específicos a la implementación.

Las asociaciones representan relaciones entre instancias de clases. Por ejemplo, una persona trabaja para una empresa; la empresa tiene varias oficinas. La figura 23 ilustra este ejemplo:

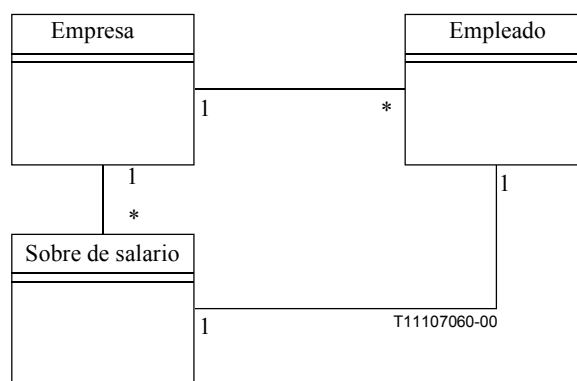


Figura 23/Q.65 – Asociaciones de instancias de clases

La figura 23 muestra la relación entre una empresa, sus empleados y los sobres que contienen sus salarios. En esta relación una empresa tiene muchos empleados, y un empleado tiene un solo sobre de salario. Alternativamente, un sobre de salario se adscribe a un empleado, y un empleado tiene un solo empleador.

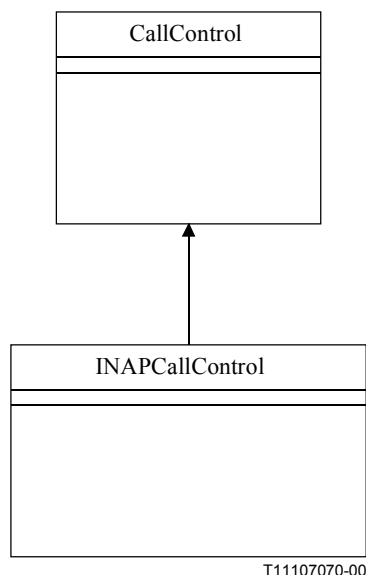


Figura 24/Q.65 – Uso de subtipos o subclases

En la figura 24 se explica el uso de subtipos o subclases. Aquí el servicio genérico es CallControl (control de llamada), mientras que "INAPCallControl" es un tipo de control de llamada específico a los sistemas asociados a protocolo de aplicación de red inteligente (INAP, *intelligent network application protocol*).

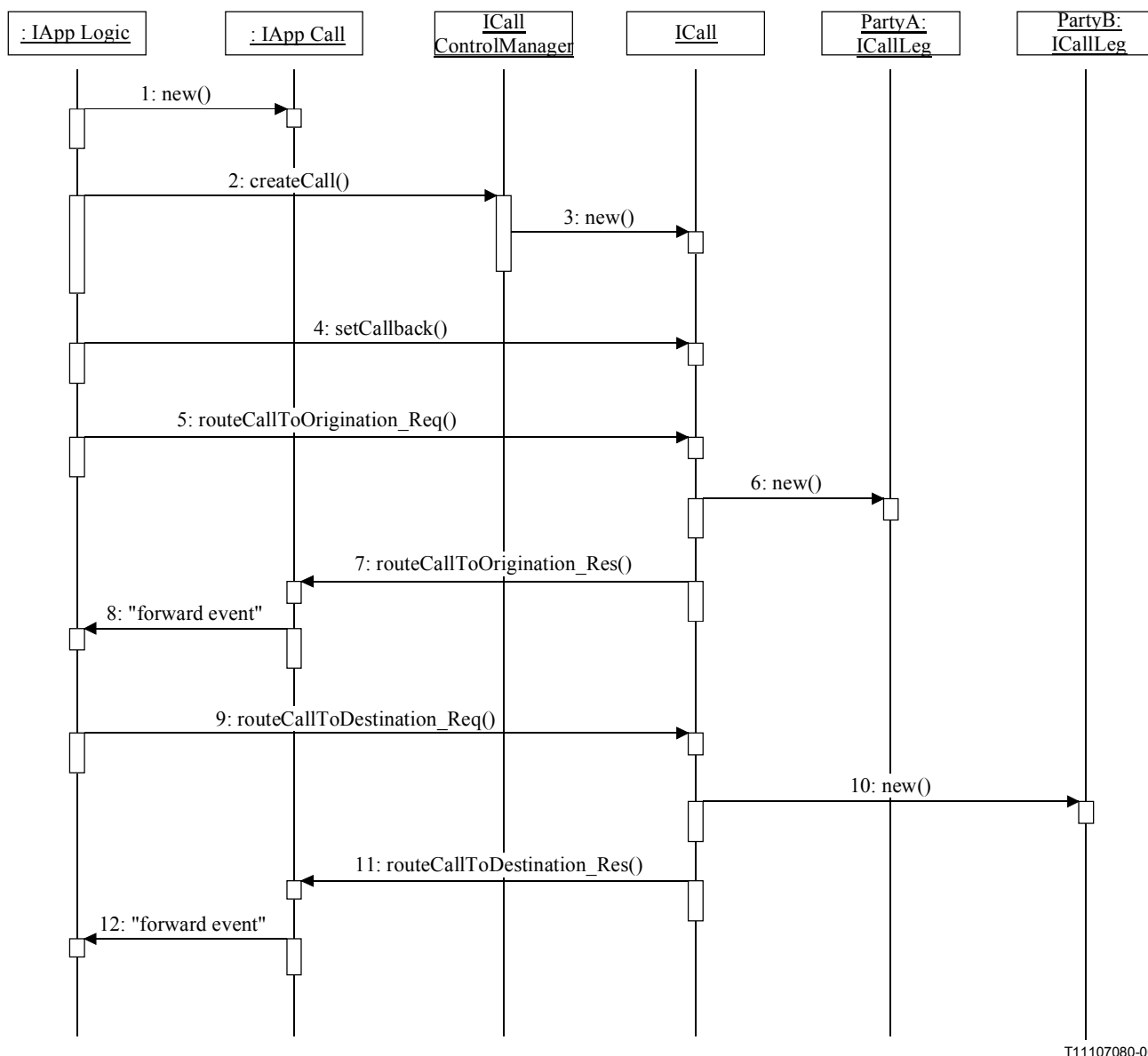
3.2.3 Diagramas de secuencias de interfaz (opcional)

La utilización de los diagramas de secuencias de interfaz es opcional. Los diagramas de secuencias de interfaz son parecidos, por su naturaleza, a los "diagramas de flujo de la información" de 2.3.1. Sin embargo, los diagramas de secuencias de interfaz pueden también indicar flujos de llamadas a API (invocaciones de método), [véase la explicación en 3.3], representándose cualquiera de los lados de la interfaz. Como es posible que las llamadas a API dentro de cada sistema estén patentadas, la utilización de tales diagramas dentro de una Recomendación puede estar restringida u omitida.

Los diagramas de secuencias de interfaz representan en forma gráfica las invocaciones de método (llamadas a API) que atraviesan una determinada interfaz, realizando así el ordenamiento en el tiempo de los mensajes. Desde el punto de vista O-O, representan a las invocaciones de método entre distintas clases Interfaz. Con ello puede que algunas de estas invocaciones de método no fluyan realmente a través de una interfaz sino que sean también internas a algún sistema particular. Aún así, por cuestiones de compleción se deberían mostrar todas las invocaciones de método.

Como muestra la figura 25, un diagrama de secuencias se forma colocando las clases Interfaz (objetos) que participan en la interacción en la parte superior del diagrama, sobre el eje de abscisas. En un caso típico se colocará el objeto que inicia la interacción a la izquierda, y los objetos de subordinación creciente a la derecha. Ahora se deben colocar los mensajes que envían y reciben estos objetos sobre el eje de ordenadas, a lo largo del cual el tiempo crece desde arriba hacia abajo. Esto le permite al lector distinguir claramente cómo fluye el control en función del tiempo.

Los diagramas de secuencias tienen dos características que los distinguen de los diagramas de colaboración. En primer lugar, la recta de vida del objeto. Ésta es la recta vertical que representa la existencia de un objeto a lo largo del tiempo. Se pueden crear objetos durante la interacción, y su existencia se inicia con la recepción del mensaje "New" (nuevo). Análogamente, se pueden eliminar objetos.



T11107080-00

Figura 25/Q.65 – Ejemplo de diagrama de secuencia de interfaz

El escenario previamente descrito representa una "llamada iniciada por terceros", la cual se describe más detalladamente en el apéndice III. En este ejemplo, la interfaz de hecho está entre los objetos IappCall e Icall ControlManager. Por lo tanto, las invocaciones de método 2, 4, 5, 7, 9 y 11 se manifiestan de hecho como llamadas a API a través de la interfaz física. Las otras invocaciones de método son internas a los sistemas de ambos lados de la interfaz, y por lo tanto no siempre es necesario mostrar el resultado del objeto "de destino" ya que puede que ello no afecte a la interfaz misma. Este ejemplo nos permite ver que la API necesaria para implementar el servicio/aplicación "establecimiento de la llamada por terceros" consiste en las llamadas a API 2, 4, 5, 7, 9 y 11.

A medida que se examinen más clases interfaz, la especificación de API crecerá.

3.3 Paso alternativo 4

Descripciones de API en IDL

Después de estudiados los dos pasos previos y habiendo identificado las clases interfaz que intervienen en los servicios/aplicaciones que se necesitan en su interfaz, la próxima etapa consistirá en describir detalladamente las invocaciones de métodos. Se debe recordar que las invocaciones de

método son los mensajes enviados desde una clase interfaz a otra; es por ello que cada clase Interfaz u objeto necesita saber qué información es probable que reciba. Por lo tanto, las invocaciones de método o llamadas a API deben ser descritas de una manera previamente acordada; en caso de que las llamadas a API sean adoptadas como estándar, cada sistema sabrá sobre qué información deberá actuar. La llamada a API se describe utilizando un formato acordado. Esta Recomendación recomienda el uso de un lenguaje genérico de descripción de interfaz (IDL, *generic interface description language*) para describir el API. Las ventajas de usar un formato tal es que esta descripción de interfaz puede ser rápidamente convertida a cualquier solución específica de una implementación, como por ejemplo CORBA, INAP o JAVA.

El IDL detallará cada una de las invocaciones de método y describirá los parámetros de datos que transporta dentro suyo. A continuación un ejemplo:

```
routeCallToDestination_Req(callSessionID : in TSessionID, responseRequested : in
  TCallResponseRequest, targetAddress : in TAddress, originatingAddress : in TAddress,
  originalDestinationAddress : in TAddress, redirectingAddress : in TAddress, applInfo :
  in TCallAppInfoArr) : TResult
```

En este ejemplo, el nombre de la llamada a API aparece al principio (*routeCallToDestination_Req*). Los parámetros alternativos aparecen en cursiva, para ver la descomposición del método. Por ejemplo, el parámetro *callSessionID : in TsessionID* especifica el ID de sesión de llamada de la llamada. Esto significa que el ID de la sesión de llamada será especificado como *: in TsessionID*, y entonces se encontrará el ID de llamada definido como el "tipo" estándar *sessionID*. Por lo tanto, la primera parte del parámetro explica qué se lleva en el método y la segunda parte, tras el símbolo ":", el formato en que se lleva. Los detalles exactos de dicho formato no son tema de esta Recomendación.

Para completar el proceso de modelado, se debe volver al paso 5 en 2.5.

APÉNDICE I

Formato y estructura de una descripción de la etapa 2 utilizando la metodología funcional unificada

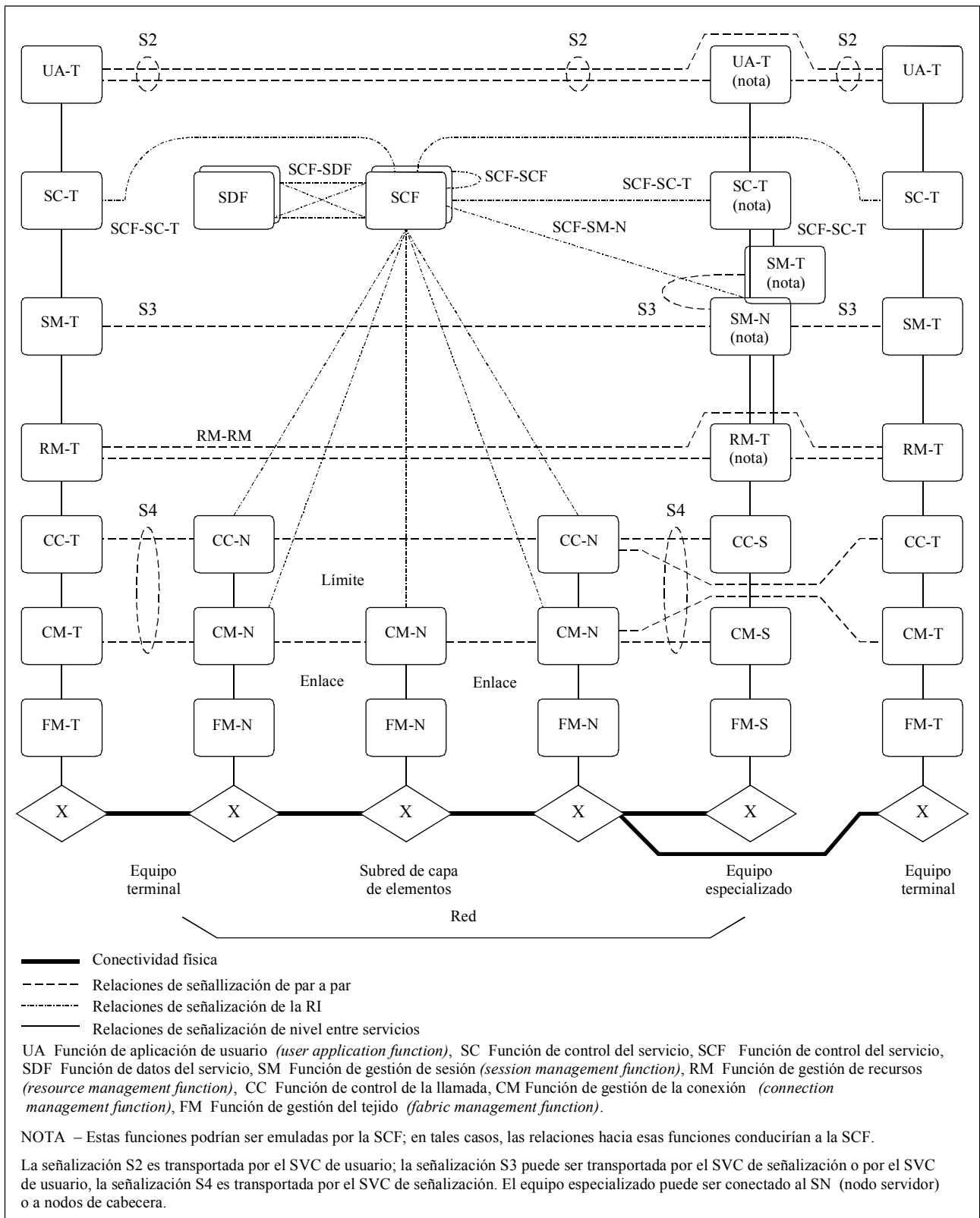
- 1 Alcance
- 2 Referencias normativas
- 3 Definiciones
- 4 Símbolos y abreviaturas
- 5 Descripción
- 6 Elaboración del modelo funcional
 - 6.1 Descripción del modelo funcional y relación con el servicio básico
 - 6.2 Descripción de las entidades funcionales
- 7 Definiciones de características de servicio basadas en los SIB
- 8 Flujos de información
 - 8.1 Diagramas de flujo de información

- 8.2 Definición de los distintos flujos de información
- 8.2.1 Relación r1
- 8.2.1.1 Contenido del flujo de información
- 8.2.1.x Contenido del flujo de información
- 9 Acciones de las entidades funcionales
- 10 Diagramas SDL de las entidades funcionales
- 11 Atribución de entidades funcionales a emplazamientos físicos (escenarios)

APÉNDICE II

Arquitectura funcional – Evolución Q.65

Este diagrama presenta un modelo de entidad funcional sin referencia a ninguna realización o arquitectura física. La entidad "CM-N" mostrada corresponde justamente al componente de control de conexión en tiempo real del NML mostrado en la arquitectura funcional unificada. La entidad FM-N mostrada corresponde justamente al recurso en tiempo real y al control del tejido del EML mostrado en la arquitectura funcional unificada.



T1183760-97

Figura II.1/Q.65 – Modelo funcional unificado

APÉNDICE III

Ejemplos de clases interfaz formadas a partir de requerimientos de servicio/aplicación

III.1 Introducción

Los siguientes ejemplos de clases interfaz se presentan aquí para ayudar al lector a definir un sistema. No deben considerarse como definitivos, y sólo deben usarse como orientación.

La metodología explicada en la cláusula 3 se usa aquí para ver cómo se elabora la API apropiada para una determinada interfaz. Describiremos un tipo de servicio/aplicación. Se describirá la aplicación, y se identificarán las resultantes clases interfaz. También se poblará cada una de las clases interfaz con los flujos de información (invocaciones de método) que se identificarán.

III.2 Ejemplo de servicio/aplicación

III.2.1 Llamada iniciada por terceros

En este ejemplo supondremos que un proveedor de servicios externo requiere una llamada entre dos partes, A y B. Para ello el proveedor de servicios debe comunicar esto a un operador de red, probablemente la función control de servicios (SCF, *service control function*). Para que esto sea posible se deben identificar las clases interfaz que tienen la responsabilidad de comunicar estas necesidades. Analizando el servicio se verifica que es necesario poder crear dos tramos distintos, establecer una llamada entre ellos y luego poder gestionar la llamada. El primer requisito, es por tanto proporcionar un "**Call Control Manager**" (gestor de control de llamadas). Esta clase Interfaz tendrá a su cargo la creación de llamadas simples o múltiples. Como puede gestionar muchas llamadas, será necesario disponer de una clase interfaz específica para cada llamada en particular, para lo cual es indispensable proporcionar una clase interfaz "**Call Control**" (control de llamada). Como cada llamada tendrá Call Legs (tramos de llamada) asociados y como cada tramo será independiente, se debe proveer una clase Interfaz "**Call Leg**" para cada tramo. Es decir, que por ahora el servicio ya tiene:

- Call Control Manager (gestor de control de llamadas)
- Call Control (control de llamada)
- Call Leg[A party] (tramo de llamada [parte A])
- Call Leg[B party] (tramo de llamada [parte B])

Cada una de estas clases interfaz, como se describió previamente en la cláusula 3, se implementará de a pares: una se ejecutará dentro de la aplicación y la otra, por así decirlo, se ejecutará en la red. La terminología usada para describir esto es:

Clases de interfaz de aplicación	Clases interfaz lado red
IAppCallControlManager	IcallControlManager
IAppCall	Icall
PartyA: IAppCallLeg	PartyA: IcallLeg
PartyB: IAppCallLeg	PartyB: IcallLeg

Se observará que la "I" que precede al nombre indica que aquí se está refiriendo a "clase interfaz" y no a una clase Objeto. Las clases interfaz carecen de atributos asociados -como sí las tienen las clases objeto- por lo cual las clases objeto son específicas a una implementación.

Llegado a este punto, será conveniente aplicar diagramas de clase. Esto es sumamente importante desde la perspectiva del sistema.

Anteriormente se analizó el concepto de paquete, y a continuación será objeto de un estudio más a fondo.

El "paquete" es importante cuando la "interface" (interfaz) se dispone en partes conexas. Por ejemplo, se podría desear agrupar todos los aspectos de Call Control (gestión de llamadas) en un único paquete, permitiendo así replicar estas clases interfaz en distintas interfaces sin tener que pasar una y otra vez por la misma captura de requerimientos. En la siguiente cláusula se examinará un diagrama de clase de servicio de control de llamada, y se mostrará la relación entre paquetes y clases interfaz.

III.3 Diagramas de clase

III.3.1 Diagrama de clase de servicio genérico de control de llamada

El paquete de aplicación de servicio genérico de control de llamada de la figura III.1, rotulado *IAppGCCS* (*interface application generic call control service*) consiste en una interfaz *IAppCallControlManager*, cero o más interfaces *IAppCall* y cero o más interfaces *IAppCallLeg*.

El paquete de servicio genérico de control de llamada de la figura III.1, rotulado *IGCCS*, consiste en una interfaz *ICallControlManager*, cero o más interfaces *ICall* y cero o más interfaces *ICallLeg*.



Figura III.1/Q.65 – Paquetes de servicio genérico de control de llamada

El diagrama de clase de la figura III.2 muestra las interfaces que conforman el paquete aplicación de servicio genérico de control de llamada y el paquete de servicio genérico de control de llamada. La comunicación entre estos paquetes se hace a través de los canales *+uses the ICallControlManager* ("usa el ICallControlManager"), *+uses the ICall* ("usa el ICall") y *+uses the ICallLeg* ("usa el ICallLeg").

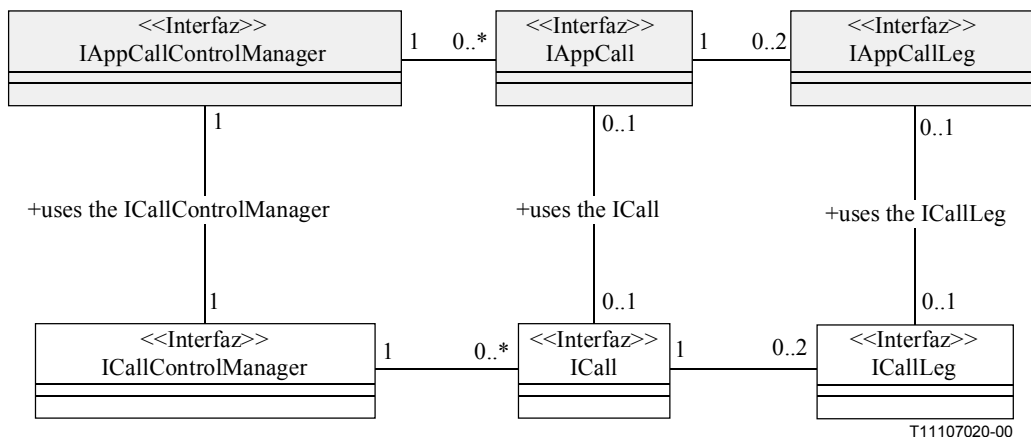


Figura III.2/Q.65 – Diagrama de clase de servicio genérico de control de llamada

III.3.2 Invocaciones de método

Llegados a este punto, es necesario poblar las clases con operaciones o, como se las conoce, "Invocaciones de método". Son estos mensajes que se pueden enviar a la clase para invocar algún resultado. En primer lugar se examinará el ICallControlManager.

III.3.2.1 Invocaciones de método ICallControlManager

En primer lugar, el cliente deberá indicar a la red que necesita establecer -"crear"- una llamada; el primer método será, por lo tanto:

- "createCall"

Por ahora es probable que con esto sea suficiente. Sin embargo, se ha de tener en cuenta que a medida que examinemos más servicios/aplicaciones, se identificarán nuevos métodos.

A continuación se examinará la clase de interfaz Call.

III.3.2.2 Invocaciones de método ICall

Convendría utilizar los siguientes métodos:

- routeCallToDestination_Req
- routeCallToOrigination_Req

Por el momento es probable que sea suficiente con estos métodos. Por último, se examinará CallLeg.

III.3.2.3 Invocaciones de método ICallLeg

Convendría utilizar los siguientes métodos:

- routeCallLegToAddress
- releaseCallLeg

Habiendo dado una definición provisoria de las anteriores clases interfaz, es necesario ahora describir en detalle el lenguaje de descripción de interfaz (IDL, *interface description language*) que implementa las invocaciones de método.

III.3.3 IDL

Sólo se examinarán en detalle, como ejemplo, dos de las clases interfaz. Este proceso debe repetirse a su turno para cada una de las clases.

Nuevamente, las siguientes clases interfaz son sólo ejemplos:

III.3.3.1 IDL ICallControlManager

Clase interfaz

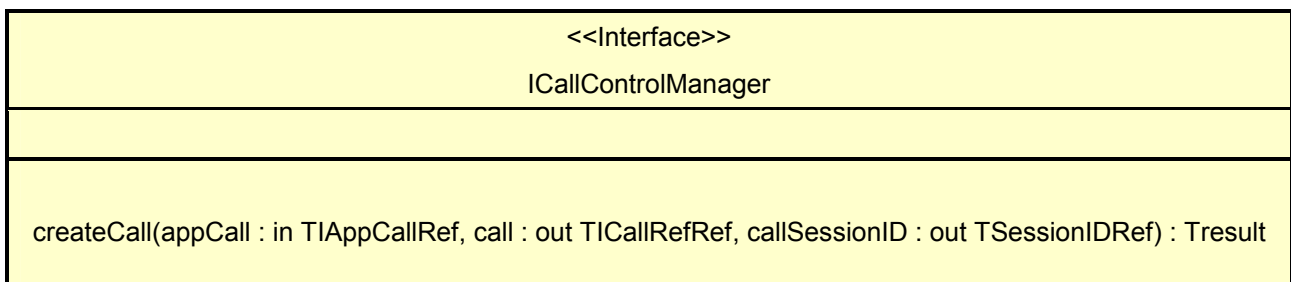


Figura III.3/Q.65

A continuación se describe el método createCall y los parámetros asociados.

Método

createCall()

Este método se usa para crear un nuevo objeto llamada.

Parámetros

appCall: in TAppCallRef

Especifica la interfaz de aplicación para llamadas en sentido inverso desde la llamada creada.

call: out TCallRefRef

Especifica la referencia de interfaz de la llamada creada.

callSessionID: out TSessionIDRef

Especifica la ID de la sesión de llamada de la llamada creada.

III.3.3.2 ICall IDL

La figura III.4 es un ejemplo de la clase interfaz "ICall"

Clase interfaz

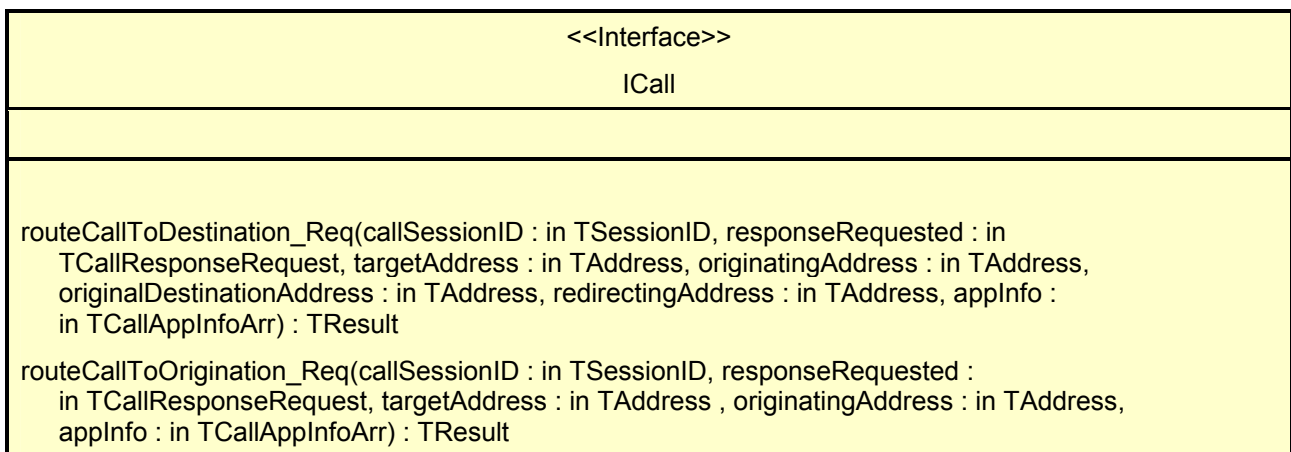


Figura III.4/Q.65

A continuación se describen los métodos routeCallToDestination_Req y routeCallToOrigination_Req y los parámetros asociados.

Método

routeCallToDestination_Req()

Este método asíncrono requiere que se encamine la llamada (y las partes inherentemente ligadas a la misma) al destinatario a través de un tramo de llamada pasivo (que se crea implícitamente).

Parámetros

callSessionID: in TSessionID

Especifica la ID de sesión de llamada de la llamada.

responseRequested: in TCallResponseRequest

Especifica el conjunto de eventos observados que resultarán en que se genere una **routeCallToDestination_Res.**

targetAddress: in TAddress

Especifica el destinatario a quien se debe encaminar la llamada.

originatingAddress: in TAddress

Especifica la dirección de la parte originadora (la parte que llama).

originalDestinationAddress: in TAddress

Especifica la dirección de destino original de la llamada.

redirectingAddress: in TAddress

Especifica la última dirección desde la cual se reencaminó la llamada.

appInfo: in TCallAppInfoArr

Especifica la información relativa a la aplicación pertinente a la llamada (tal como método de alerta, tipo de teleservicio, identidades de servicio, indicadores de interacción).

Método

routeCallToOrigination_Req()

Este método asíncrono requiere que se encamine la llamada a la primera parte de la llamada a través de un tramo de llamada de control (el cual es creado implícitamente). El objeto llamada ya debe haber sido creado.

Parámetros

callSessionID: in TSessionID

Especifica el ID de la sesión de llamada de la llamada.

responseRequested: in TCallResponseRequest

Especifica el conjunto de eventos observados que resultarán en que se genere una **routeCallToOrigination_Res()**.

targetAddress: in TAddress

Especifica el originador a quien se debe encaminar la llamada.

originatingAddress: in TAddress

Especifica la dirección de la parte originadora (quien llama)

AppInfo: in TCallAppInfoArr

Especifica información relativa a la aplicación y pertinente a la llamada (tal como método de alerta, tipo de teleservicio, identidades de servicio, indicadores de interacción).

III.4 ¿A dónde ir ahora?

En este apéndice III se ha explicado cómo se usa la herramienta UML para especificar servicios/aplicaciones. Así, se ha llegado a un punto similar al del método alternativo (usando los SIB), en el cual se debe poner en correspondencia los resultados con un protocolo. Al igual que en el enfoque de SIB, se debe ahora ir directamente a la SDL para definir el sistema sobre el cual se invocan estos métodos; aquí hay dos opciones posibles:

- 1) se pueden enviar los resultados anteriores a la máquina SDL y generar el SDL correspondiente; o bien
- 2) se puede especificar el IDL de una manera específica a una implementación, tal como decidir si usar una solución CORBA u otra. Con esta primera correspondencia directa, es aún posible poner en correspondencia -mediante la herramienta apropiada- la solución IDL CORBA directamente en SDL.

Estas dos opciones no son tema de esta Recomendación.

SERIES DE RECOMENDACIONES DEL UIT-T

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