



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**Q.2630.1**

**Anexo B**  
(03/2001)

**SERIE Q: CONMUTACIÓN Y SEÑALIZACIÓN**

Red digital de servicios integrados de banda ancha (RDSI-BA) – Aspectos comunes de los protocolos de aplicación de la RDSI-BA para la señalización de acceso, la señalización de red y el interfuncionamiento

---

Protocolo de señalización de la capa de adaptación del modo transferencia asíncrono tipo 2 – Conjunto de capacidades 1

**Anexo B: Definición en SDL del conjunto de capacidades 1 del protocolo de señalización de la capa de adaptación del modo de transferencia asíncrono tipo 2**

Recomendación UIT-T Q.2630.1 – Anexo B

(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
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.799
INTERFAZ Q3	Q.800–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
ESPECIFICACIONES DE LA SEÑALIZACIÓN RELACIONADA CON EL CONTROL DE LLAMADA INDEPENDIENTE DEL PORTADOR	Q.1900–Q.1999
RED DIGITAL DE SERVICIOS INTEGRADOS DE BANDA ANCHA (RDSI-BA)	Q.2000–Q.2999
Aspectos generales	Q.2000–Q.2099
Capa de adaptación del modo de transferencia asíncrono de señalización	Q.2100–Q.2199
Protocolos de red de señalización	Q.2200–Q.2299
<b>Aspectos comunes de los protocolos de aplicación de la RDSI-BA para la señalización de acceso, la señalización de red y el interfuncionamiento</b>	<b>Q.2600–Q.2699</b>
Protocolos de aplicación de la RDSI-BA para señalización de red	Q.2700–Q.2899
Protocolos de aplicación de la RDSI-BA para señalización de acceso	Q.2900–Q.2999

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

## **Recomendación UIT-T Q.2630.1**

### **Protocolo de señalización de la capa de adaptación del modo transferencia asíncrono tipo 2 – Conjunto de capacidades 1**

#### **ANEXO B**

#### **Definición en SDL del conjunto de capacidades 1 del protocolo de señalización de la capa de adaptación del modo de transferencia asíncrono tipo 2**

#### **Resumen**

Este anexo contiene la definición SDL del conjunto de capacidades 1 del protocolo de señalización de la capa de adaptación del modo de transferencia asíncrono tipo 2 para la Recomendación UIT-T Q.2630.1. Los diagramas SDL están disponibles sólo en formato electrónico.

#### **Orígenes**

El anexo B a la Recomendación UIT-T Q.2630.1, preparado por la Comisión de Estudio 11 (2001-2004) del UIT-T, fue aprobado por el procedimiento de la Resolución 1 de la AMNT el 1 de marzo de 2001.

## PREFACIO

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

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

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

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

## NOTA

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

## PROPIEDAD INTELECTUAL

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

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

© UIT 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

	<b>Página</b>
B.1 Introducción .....	1
B.2 El diagrama del sistema SDL.....	1
B.3 El diagrama de bloques del SDL.....	1
B.4 Especificación de la función nodal en SDL.....	7
B.4.1 Introducción .....	7
B.4.2 Diagramas SDL de la función nodal 1 .....	7
B.4.3 Procedimientos en la función nodal.....	14
B.4.4 Estructura de datos de los mensajes y parámetros de señalización AAL tipo 2 .....	15
B.5 Diagramas SDL de las entidades de protocolo .....	44
B.5.1 Introducción .....	44
B.5.2 Diagramas SDL de los procedimientos de protocolo saliente, entrante y de mantenimiento .....	44



## **Recomendación UIT-T Q.2630.1**

### **Protocolo de señalización de la capa de adaptación del modo transferencia asíncrono tipo 2 – Conjunto de capacidades 1**

#### **ANEXO B**

##### **Definición en SDL del conjunto de capacidades 1 del protocolo de señalización de la capa de adaptación del modo de transferencia asíncrono tipo 2**

Las definiciones en SDL pueden ser más detalladas que las definiciones en lenguaje corriente de la cláusula 8. No obstante, de haber diferencias de carácter técnico entre este anexo y la cláusula 8, prevalecerán las definiciones que figuran en dicha cláusula.

#### **B.1 Introducción**

Las definiciones en SDL del protocolo de señalización AAL tipo 2 que figuran en la presente Recomendación dependen del sistema SDL y de los diagramas de bloques definidos en el presente anexo.

Las definiciones en SDL del presente anexo suponen la existencia de un evento único en un tiempo dado, por consiguiente no se consideran las condiciones de competencia dentro de la entidad de señalización AAL de tipo 2. La resolución de las colisiones y condiciones de competencia depende de la implementación.

#### **B.2 El diagrama del sistema SDL**

El diagrama del sistema SDL se representa en la figura B.1.

#### **B.3 El diagrama de bloques del SDL**

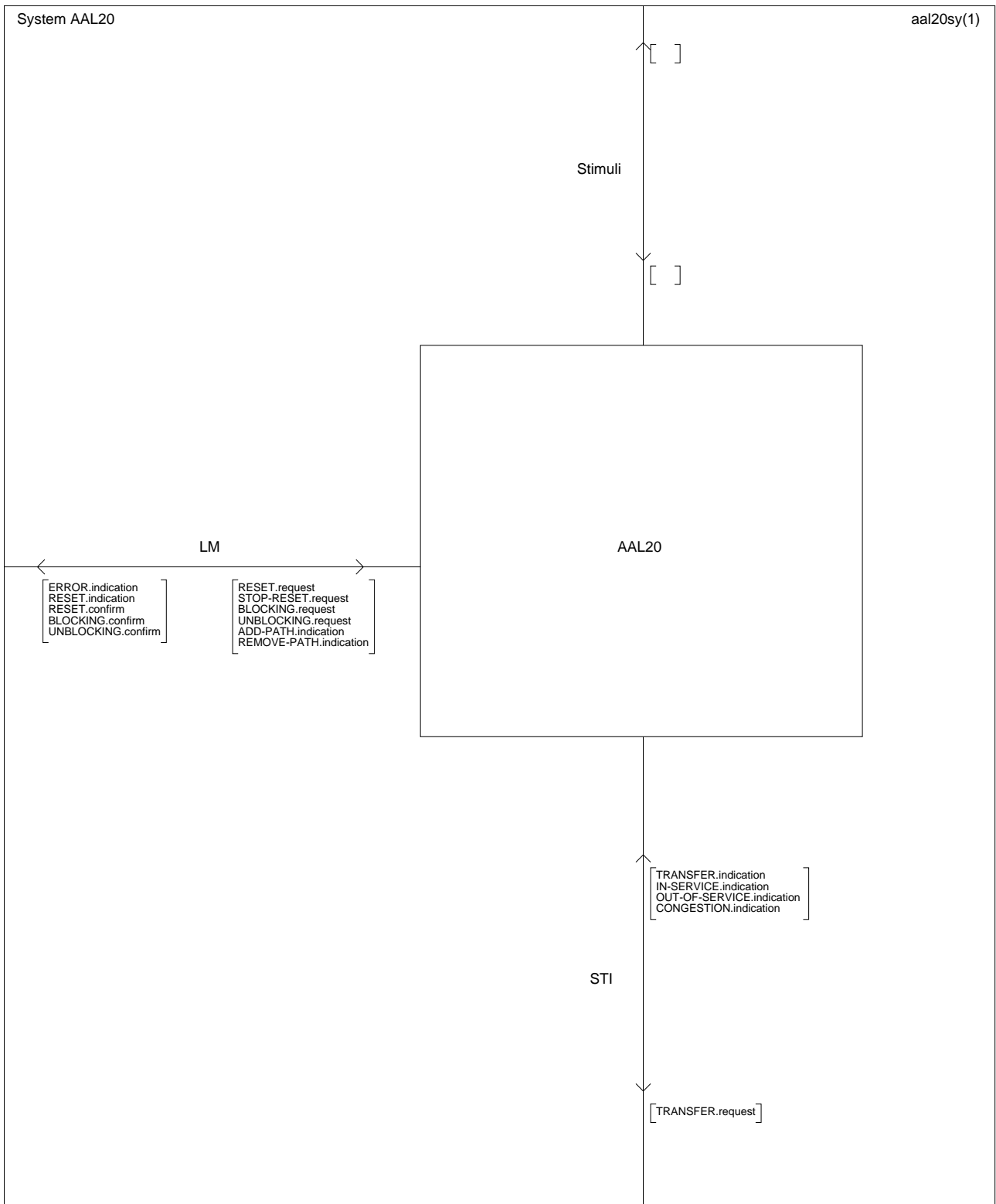
Los diagramas de bloques del SDL se representan en la figura B.2 (partes 1 a 4 de 4).

NOTA 1 – Ni el bloque USER ni su correspondiente proceso USER (que no se muestra) forman parte de la entidad de señalización AAL tipo 2. Sólo se utilizan para indicar distintas entidades de usuario servido.

NOTA 2 – Los procedimientos situados en el proceso NodalF2 y llamados por el proceso NodalF1 no se explican con detalle en este anexo.

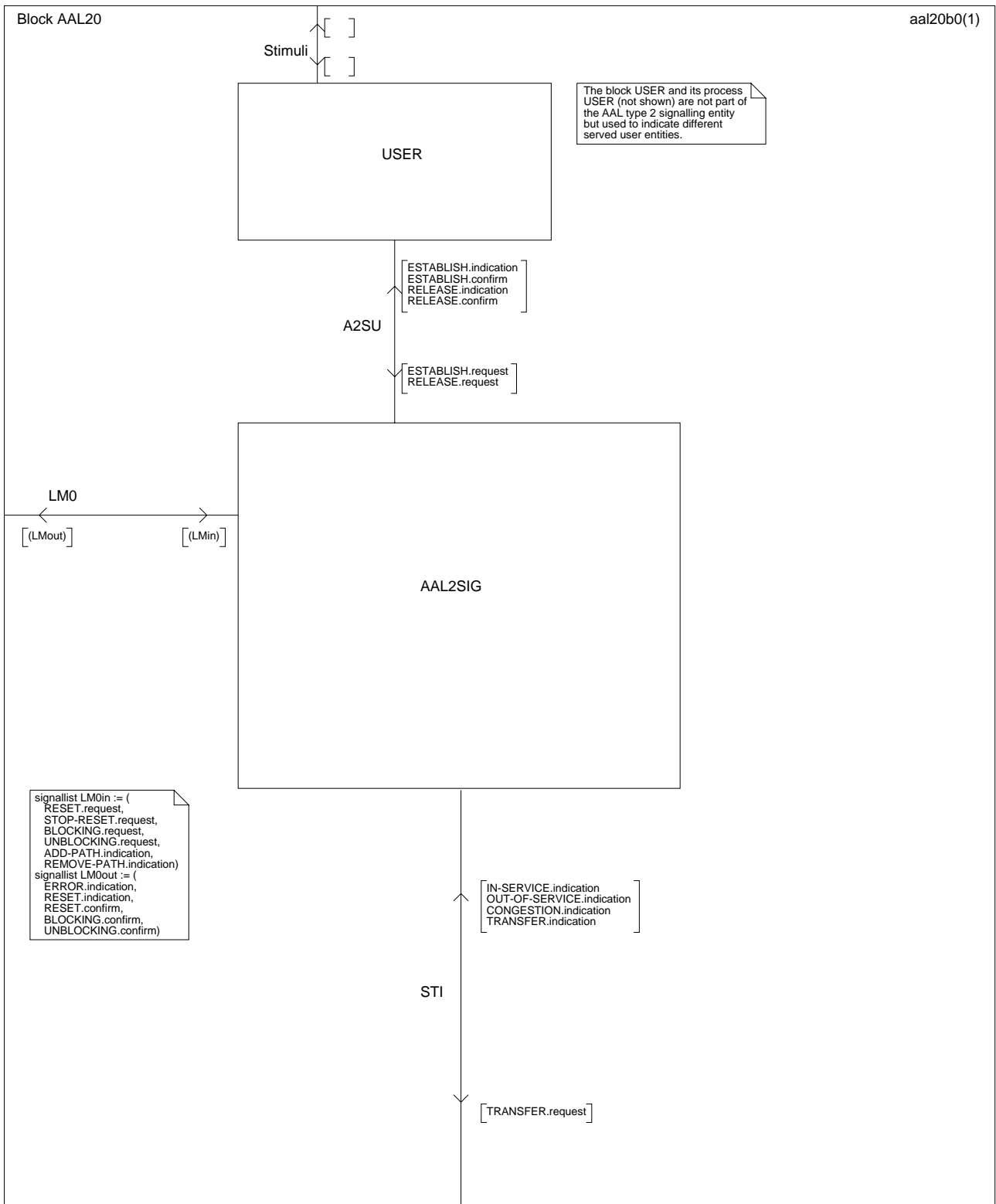
NOTA 3 – Las llamadas del proceso NodalF1 a los procedimientos situados en el proceso NodalF2 provocan un intercambio de señales implícitas entre los procesos NodalF1 y NodalF2.

NOTA 4 – Hay una entidad STI por convertidor de transporte de señalización. Estos convertidores son conocidos por la función nodal 2 con su ProcessID (SDL). No se muestran en los diagramas SDL de este anexo la adición ni la supresión de las relaciones de señalización inherentes a la creación y destrucción de los procesos STI y STC.

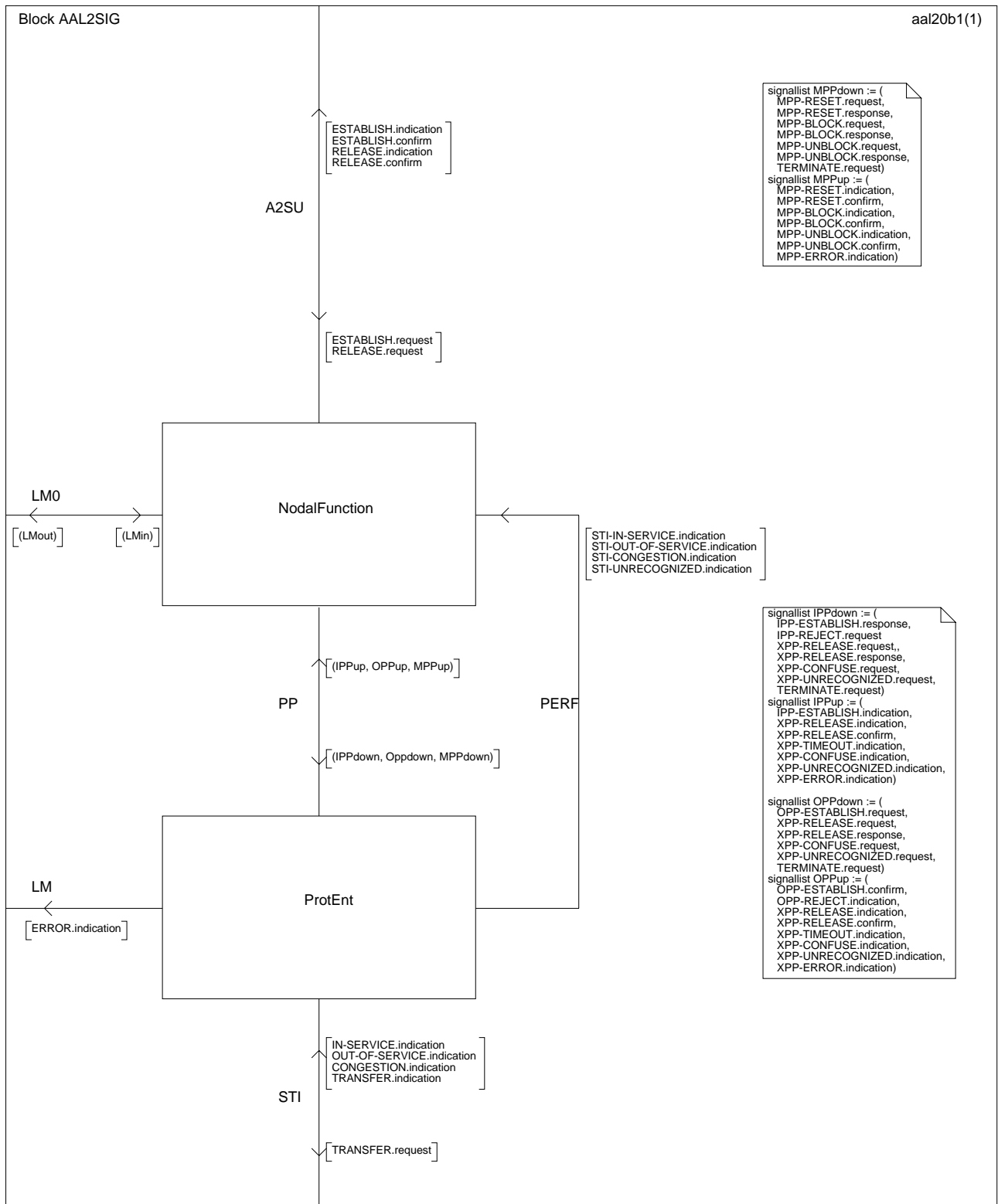


**Figura B.1/Q.2630.1 – Sistema SDL para la entidad de señalización AAL tipo 2**

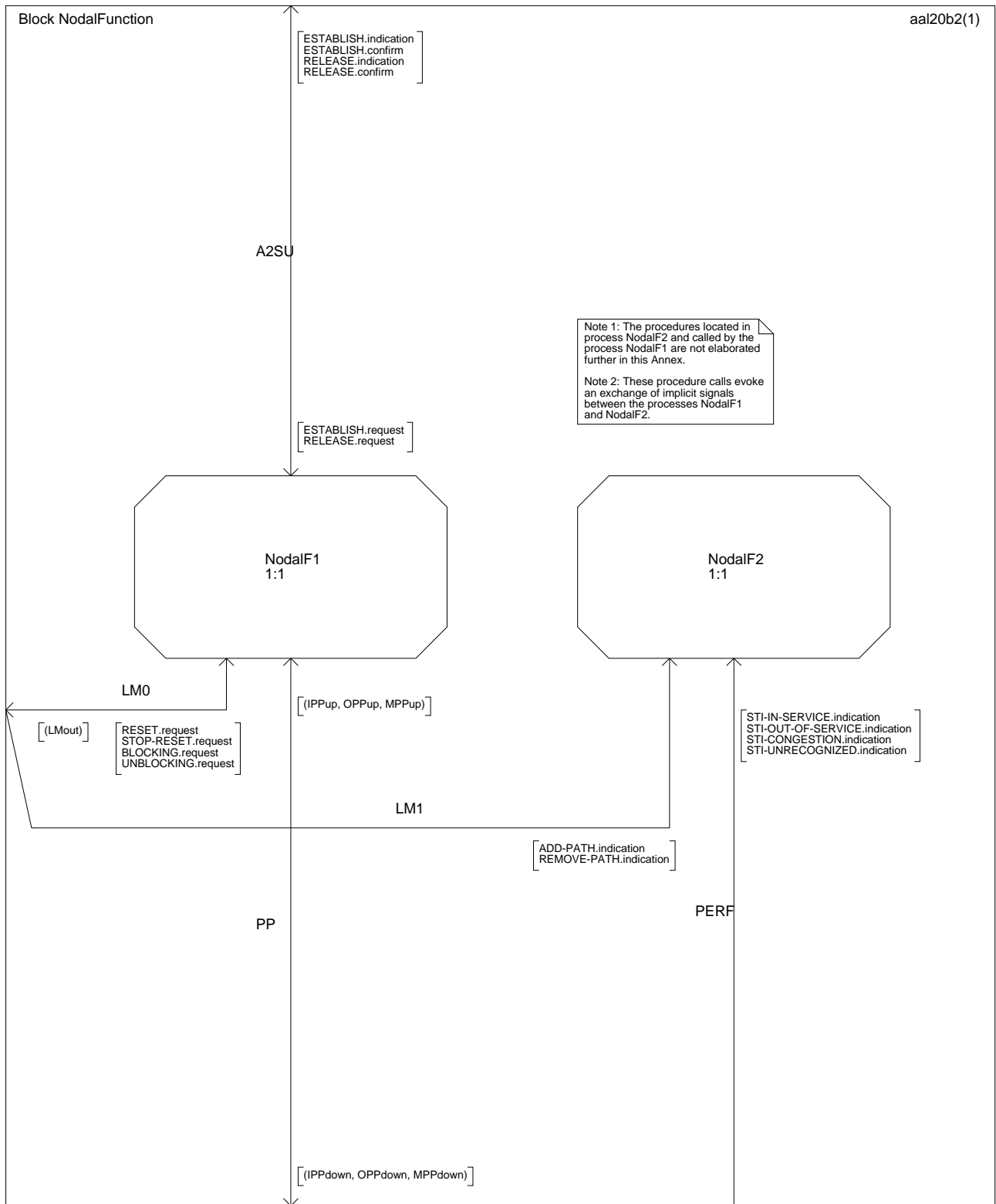




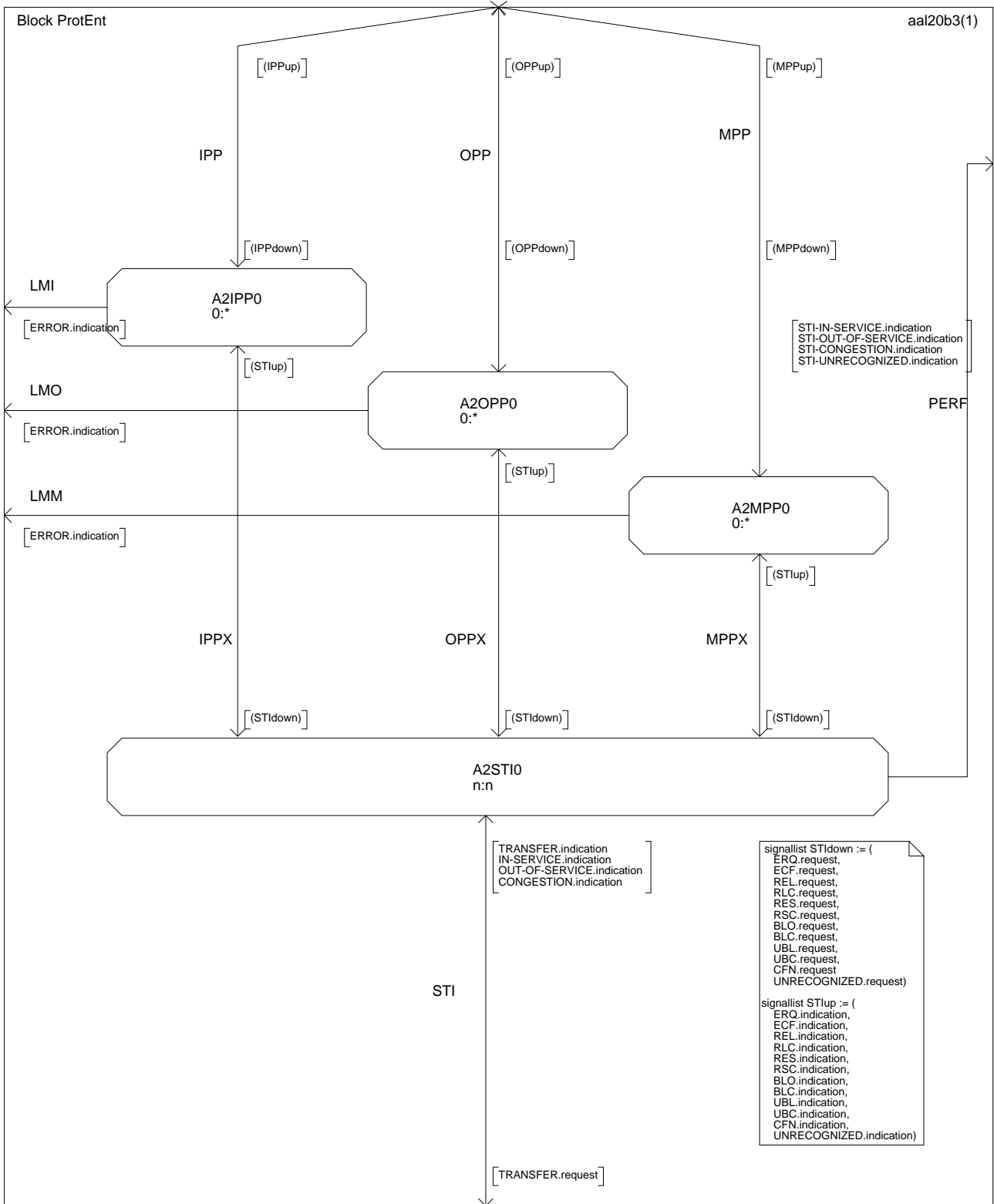
**Figura B.2/Q.2630.1 – Estructura de bloque SDL para la entidad de señalización AAL tipo 2 (parte 1 de 4)**



**Figura B.2/Q.2630.1 – Estructura de bloque SDL para la entidad de señalización AAL tipo 2 (parte 2 de 4)**



**Figura B.2/Q.2630.1 – Estructura de bloque SDL para la entidad de señalización AAL tipo 2 (parte 3 de 4)**



**Figura B.2/Q.2630.1 – Estructura de bloque SDL para la entidad de señalización AAL tipo 2 (parte 4 de 4)**

## B.4 Especificación de la función nodal en SDL

### B.4.1 Introducción

La figura B.2 (parte 3 de 4) muestra la función nodal separada en:

a) *Función nodal 1*

Esta entidad contiene todas las funcionalidades detalladas en 8.2 y, por lo tanto, se puede especificar con precisión en diagramas SDL.

b) *Función nodal 2*

Esta entidad contiene todas las funcionalidades que tengan partes específicas de la aplicación. En 8.2 sólo se detallan los resultados (aunque no sus mecanismos internos). Estas partes tampoco se detallan en este anexo; sólo se documenta el resultado esperado.

Por lo tanto, esta cláusula define la función nodal 1 (véase B.4.2) y los procedimientos de la función nodal 2 (véase B.4.3). Estos últimos, por su naturaleza, no se especifican en SDL; sin embargo, hay que comprender la funcionalidad esperada de estos procedimientos para completar la definición de los diagramas SDL de la entidad de señalización AAL tipo 2.

No se definen los procesos USER, es decir los usuarios servidos por la AAL tipo 2 ni tampoco la interacción de estos procesos con el entorno (por ejemplo para estímulos de simulación). Se supone que existe un proceso USER independiente para cada conexión AAL tipo 2 (en ambos extremos de la conexión) y que se utiliza su identidad (proceso SDL) para encaminar las señales.

NOTA – Los procesos de las entidades de protocolo se describen en 8.3.

### B.4.2 Diagramas SDL de la función nodal 1

#### B.4.2.1 Estructuras de datos

En el modelo SDL, la entidad de señalización AAL tipo 2 mantiene un registro del tipo "CRec" por cada conexión AAL del tipo 2 desde el instante en que se establece la conexión hasta que se libera. Los términos "precedente" y "siguiente" en el texto a continuación se refieren al sentido del establecimiento de la conexión.

Cuando se busca este registro (con el identificador de ejemplar de la entidad del procedimiento de protocolo), se organiza el mismo de tal manera que la parte "entrante" se refiere al enlace desde el que se ha recibido el mensaje; esto se refleja en el valor de estado descrito en el cuadro B.1.

**Cuadro B.1/Q.2630.1 – Valores de estado para los registros CRec**

Descripción	Valor de estado	Enlace descrito por la parte entrante	Enlace descrito por la parte saliente
Pendiente de establecimiento	2	precedente	subsiguiente
	3	subsiguiente	precedente
Conexión establecida	4	precedente	subsiguiente
	5	subsiguiente	precedente
Pendiente de liberación	6	precedente	subsiguiente
	7	subsiguiente	precedente

La estructura del registro del tipo "CRec" se define en el fragmento ASN.1 a continuación:

```

CRec ::= SEQUENCE {
    status      CRecStatus,      -- connection status
    incoming    HRec,            -- details incoming link
    outgoing    HRec }          -- details outgoing link

HRec ::= SEQUENCE {
    peer        ENUMERATED {user,remote,none},
    ppus        PID,            -- ID of protocol entity or user
    srid        PID,            -- signalling relation identifier
                                -- is ID of signalling relation
    ceid        CEID }         -- connection element identifier
    
```

En el modelo SDL, la entidad de señalización AAL tipo 2 mantiene un registro del tipo "MRec" por cada acción de mantenimiento saliente desde el instante en que se inicia la acción de mantenimiento hasta que se concluye. La estructura del MRec se define en el fragmento ASN.1 siguiente:

```

MRec ::= SEQUENCE {
    status      MRecStatus,      -- maintenance action status
    ppus        PID,            -- ID of protocol entity
    srid        PID,            -- signalling relation identifier
                                -- is ID of signalling transport entity
    ceid        CEID,          -- connection element identifier
    disp        BOOLEAN }       -- TRUE: originator is layer management
    
```

#### B.4.2.2 Primitivas entre la función nodal 1 y las entidades de protocolo

En 5.1 se define la interfaz hacia el usuario servido AAL tipo 2. En 5.3 se define la interfaz con la gestión de capa. La interfaz entre la entidad de señalización AAL tipo 2 y las entidades de protocolo AAL tipo 2 se resumen en el cuadro B.2 y se definen después del cuadro.

**Cuadro B.2/Q.2630.1 – Primitivas y parámetros intercambiados entre la entidad función nodal 1 y las entidades de protocolo**

Nombre genérico de la primitiva	Tipo			
	Petición	Indicación	Respuesta	Confirmación
IPP-ESTABLISH	No definida	ERQmsg, sri	ECFmsg	No definida
IPP-REJECT	RLCmsg	No definida	No definida	No definida
OPP-ESTABLISH	ERQmsg	No definida	No definida	ECFmsg
OPP-REJECT	Not defined	RLCmsg	No definida	No definida
XPP-RELEASE	RELmsg	RELmsg	RLCmsg	RLCmsg
XPP-TIMEOUT	Not defined	–	No definida	No definida
XPP-CONFUSE	CNFmsg	CNFmsg	No definida	No definida
XPP-UNRECOGNIZED	msg	msg	No definida	No definida
MPP-BLOCK	BLOmsg, sqc	BLOmsg, sri	BLCmsg	BLCmsg
MPP-UNBLOCK	UBLmsg	UBLmsg, sri	UBCmsg	UBCmsg
MPP-RESET	RESmsg	RESmsg, sri	RSCmsg, sqc	RSCmsg
MPP-ERROR	No definida	causa	No definida	No definida
TERMINATE	–	No definida	No definida	No definida

– Esta primitiva no tiene parámetros.

a) **IPP-ESTABLISH**

La entidad de protocolo entrante recién creada indica el mensaje de petición de establecimiento (ERQ, *establish request message*) a la función nodal 1 (con la primitiva de indicación) junto con el identificador de asociación de señalización nodal "sri". Si el establecimiento concluye con un usuario servido en este nodo o si los nodos AAL tipo 2 más allá del enlace siguiente acusan recibo del establecimiento, se envía el mensaje de confirmación de establecimiento (ECF, *establish confirm message*) a la entidad de protocolo entrante (con la primitiva respuesta) para su transmisión por el enlace precedente.

b) **IPP-REJECT**

Si no se puede completar el establecimiento en este nodo AAL tipo 2 o en cualquier nodo AAL tipo 2 más allá del enlace siguiente, se rechaza el establecimiento con un mensaje de confirmación de liberación (RLC) que se envía a la entidad de protocolo entrante (con la primitiva petición) para su transmisión por el enlace precedente.

c) **OPP-ESTABLISH**

Si se debe continuar el establecimiento en un enlace subsiguiente se envía el mensaje de petición de establecimiento (ERQ) a la entidad de protocolo saliente recién creada (con la primitiva petición). Si la entidad de protocolo saliente recibe el mensaje de confirmación de establecimiento (ECF), éste se comunica a la función nodal 1 (con la primitiva confirmación).

d) **OPP-REJECT**

Si los nodos AAL tipo 2 más allá del enlace siguiente no pueden aceptar el establecimiento, la entidad de protocolo saliente recibe un mensaje de confirmación de liberación (RLC, *release confirm message*) que se comunica a la función nodal 1 (con la primitiva indicación).

e) **XPP-RELEASE**

Se indica a la entidad de protocolo entrante o saliente que inicie los procedimientos de liberación con el mensaje de petición de liberación (REL, *release request message*), que se envía a la entidad de protocolo (con la primitiva petición). Si la entidad de protocolo entrante o saliente recibe tal mensaje de petición de liberación (REL), éste se envía a la función nodal 1 (con la primitiva indicación). Después de la recepción del mensaje de petición de liberación (REL), la función nodal libera los recursos y confirma la liberación con el mensaje de confirmación de liberación (RLC); este mensaje se envía a la entidad de protocolo (con la primitiva respuesta). Si la entidad de protocolo entrante o saliente recibe el mensaje de confirmación de liberación (RLC), éste se envía a la función nodal 1 (con la primitiva confirmación).

f) **XPP-TIMEOUT**

Hay varias etapas en las que la entidad de protocolo entrante o saliente mantiene activo un temporizador. Si tal temporizador expira, se comunica el evento a la función nodal 1 (con la primitiva indicación); no es necesario incluir parámetros.

g) **XPP-CONFUSE**

Cuando dos nodos AAL tipo 2 de distintos niveles funcionales se comunican entre sí, no toda la información transmitida por un nodo puede ser comprendida por el otro. En tal caso, el mecanismo de compatibilidad puede requerir la transmisión del mensaje confusión (CNF); este mensaje se envía a la entidad de protocolo entrante o saliente (con la primitiva petición) para su transmisión. Cuando la entidad de protocolo entrante o saliente recibe el mensaje confusión (CNF, *confuse message*), éste se envía a la función nodal 1 (con la primitiva indicación).

**h) XPP-UNRECOGNIZED**

Cuando dos nodos AAL tipo 2 de niveles funcionales distintos se comunican entre sí, no toda la información transmitida por un nodo puede ser comprendida por el otro. En tal caso, la entidad de protocolo entrante o saliente podría recibir un mensaje no reconocido; este mensaje se enviaría a la función nodal 1 (con la primitiva indicación). El mecanismo de compatibilidad puede requerir la retransmisión de este mensaje por un enlace posterior; el mensaje no reconocido se comunica a la entidad de protocolo entrante o saliente (con la primitiva petición) para su transmisión.

**i) MPP-BLOCK**

Se solicita a una entidad de protocolo de mantenimiento recién creada la transmisión del mensaje de petición de bloqueo (BLO) (con la primitiva petición). Cuando recibe el mensaje de petición de bloqueo (BLO, *block request message*), la entidad de protocolo de mantenimiento recién creada envía este mensaje a la función nodal 1 (con la primitiva indicación). El mensaje de confirmación de bloqueo (BLC, *block confirm message*) a devolver se envía a la entidad de protocolo de mantenimiento (con la primitiva respuesta) para su transmisión. Si una entidad de protocolo de mantenimiento recibe el mensaje de confirmación de bloqueo (BLC), éste se envía a la función nodal 1 (con la primitiva confirmación).

**j) MPP-UNBLOCK**

Se solicita a la entidad de protocolo de mantenimiento recién creada la transmisión del mensaje de petición de desbloqueo (UBL) (con la primitiva petición). Cuando se recibe el mensaje de petición de desbloqueo (UBL, *unlock request message*), la entidad de protocolo de mantenimiento recién creada lo envía a la función nodal 1 (con la primitiva indicación). El mensaje de confirmación de desbloqueo (UBC, *unlock confirm message*) a devolver se envía a la entidad de protocolo de mantenimiento (con la primitiva respuesta) para su transmisión. Si la entidad de protocolo de mantenimiento recibe el mensaje de confirmación de desbloqueo (UBC), éste se envía a la función nodal 1 (con la primitiva confirmación).

**k) MMP-RESET**

Se solicita a la entidad de protocolo de mantenimiento recién creada la transmisión del mensaje de petición de reiniciación (RES, *reset request message*) (con la primitiva petición). Al recibir el mensaje de petición de reiniciación (RES), la entidad de protocolo de mantenimiento recién creada lo envía a la función nodal 1 (con la primitiva indicación). El mensaje de confirmación de reiniciación (RSC, *reset confirm message*) a devolver se envía a la entidad de protocolo de mantenimiento (con la primitiva respuesta) para su transmisión. Si la entidad de protocolo de mantenimiento recibe el mensaje de confirmación de reiniciación (RSC), éste se envía a la función nodal 1 (con la primitiva confirmación).

**l) MPP-ERROR**

Los errores detectados por la entidad de protocolo de mantenimiento se comunican a la función nodal 1 (con la primitiva indicación); entre estos errores se encuentran los fines de temporización.

**m) TERMINATE**

En cualquier momento, la función nodal 1 puede terminar la entidad de protocolo de mantenimiento (con la primitiva petición); no hay que enviar parámetros.

Además de los mensajes de señalización AAL tipo 2, se envían los siguientes parámetros:

**aa) sri**

El parámetro "sri" es del tipo PId (ID del proceso SDL) y se utiliza en las primitivas indicación e indica la relación de señalización nodal.



ab) **sqc**

El parámetro "sqc" se utiliza en las primitivas.MPP-BLOCK.request y MPP-RESET.response para asegurar la integridad de la secuencia de las peticiones de bloqueo (BLO) y de los mensajes de confirmación de reiniciación (RSC, *reset confirm messages*).

ac) **cause**

El parámetro "cause" se utiliza para indicar el tipo de error señalado por la entidad de protocolo de mantenimiento.

ad) **msg**

El parámetro "msg" contiene un mensaje completo de señalización AAL tipo 2 no reconocida.

La reacción ante los eventos de la señal de entrada se describe en las partes 1 a 15 (de 25) de la figura B.3.

### B.4.2.3 Procedimientos

Los procedimientos se describen en las partes 16 a 21 (de 25) de la figura B.3.

Las funciones "**Compatibility**" (de las partes 19 y 20 de la figura B.3) y "**MsgCompatibility**" (de la parte 21) verifican la compatibilidad de los parámetros (de los mensajes reconocidos) y determinan la acción de compatibilidad para los mensajes no reconocidos, devolviendo un valor definido por la siguiente estructura ASN.1:

<b>Compat ::= SEQUENCE {</b>	<i>-- return value of procedure</i>
	<i>Compatibility</i>
<b>course</b>	<b>ENUMERATED {pass,passcnf,dcrd,dcrdcnf,release},</b>
<b>cause</b>	<b>CAUSE }</b> <i>-- cause if not "pass"</i>

course toma los siguientes valores:

- pass: Todos los parámetros se reconocen o se pueden dejar pasar; los parámetros no reconocidos que deban desecharse se eliminan del mensaje dentro de esta función.
- passcnf: El mensaje contenía parámetros no reconocidos que se han eliminado del mensaje dentro de esta función; se solicitó una notificación.
- dcrd: El mensaje contenía parámetros no reconocidos; la existencia de un parámetro no reconocido (como mínimo) obligó a descartar el mensaje completo.
- dcrdcnf: Igual que "dcrd" pero además con petición de notificación.
- release: El mensaje contenía parámetros no reconocidos; la existencia de (al menos) un parámetro no reconocido obligó a liberar la conexión.

NOTA – "MsgCompatibility" no devuelve el valor "passcnf".

La función "**LookupCRec**" (de la parte 16 de la figura B.3) busca todos los registros del tipo "CRec" para encontrar el *crec.incoming.ppus* o el *crec.outgoing.ppus* que concuerda con el parámetro de entrada; además, la parte *crec...peer* debe mostrar el valor "remoto". La búsqueda se limita a una sola coincidencia.

Si el parámetro de entrada concuerda con *crec.outgoing.ppus*, se intercambian las partes entrante y saliente del registro. Si se verifica este intercambio, también se modifica la parte de estado del registro como sigue:

- si es par (*crec.status*) entonces
  - aumentar *crec.status* en 1
- de lo contrario
  - disminuir *crec.status* en 1
- fin

El valor devuelto se puede interpretar como puntero hacia el propio registro.

La función "**LookupMRec**" (de la parte 16 de la figura B.3) busca todos los registros del tipo "MRec" para encontrar el que concuerda con el parámetro `mrec.ppus`, limitándose a una sola coincidencia. El valor devuelto se puede interpretar como puntero hacia el propio registro.

La función "**FindNextCRec**" (en la parte 16 de la figura B.3) busca todos los registros del tipo "CRec" para encontrar el siguiente que concuerde con los parámetros "ceid" y "sri". El valor devuelto se puede interpretar como puntero hacia el propio registro, a menos que no se encuentren más registros (en cuyo caso se devuelve el valor "nulo"). Antes de devolver el puntero, se ajusta el contenido del registro como se definió anteriormente para el procedimiento "LookupCRec".

La función "**FindNextPath**" (de la parte 16 de la figura B.3) busca todos los trayectos asignados de la relación de señalización indicada con el parámetro "sri". El valor devuelto es el valor de un identificador de trayecto AAL tipo 2 a menos que no se encuentren más trayectos (en cuyo caso se devuelve el valor "nulo").

La función "**GetNextParam**" (de la parte 16 de la figura B.3) analiza sintácticamente el mensaje y extrae el siguiente parámetro. El valor devuelto es el parámetro (o una referencia al mismo), a menos que no se encuentren más parámetros (en cuyo caso se retorna el valor "nulo").

La función "**GetNextField**" (de la parte 16 de la figura B.3) analiza sintácticamente el parámetro y extrae el siguiente campo. El valor devuelto es el campo (o una referencia al mismo), a menos que no existan más campos (en cuyo caso se devuelve el valor "nulo").

La función "**allocate**" y el procedimiento "**release**" (de la parte 16 de la figura B.3) actúan como reservas de un sistema de gestión de memoria que asigna y libera registros de los tipos "CRec" y "MRec".

El procedimiento "**StartReset**" (de la parte 17 de la figura B.3) llama al procedimiento "ResetConn" para liberar las conexiones AAL tipo 2 afectadas por la reiniciación. Si se reinicia más de un canal se llama al procedimiento "ResetUnblock" para poner todos los trayectos AAL tipo 2 afectados a "desbloqueados a distancia". Se construye un mensaje reiniciación (RES) y se presenta a la entidad de protocolo de gestión recién creada. Antes de la devolución, se asigna y se rellena un nuevo registro del tipo "MRec".

El procedimiento "**StartBlocking**" (de la parte 17 de la figura B.3) llama al procedimiento "BLOCKING" para registrar el bloqueo solicitado antes de construir el mensaje de bloqueo (BLO, *blocking message*) y presentar el mensaje a la entidad de protocolo de gestión recién creada. Antes de la devolución, se asigna y rellena un nuevo registro del tipo "MRec".

El procedimiento "**StartUnblocking**" (de la parte 17 de la figura B.3) prueba (llamando al procedimiento "bloqueado") si los trayectos indicados ya están desbloqueados localmente; en caso afirmativo, se emite la primitiva UNBLOCK.confirm y termina el desbloqueo. De lo contrario, se construye el mensaje desbloqueo (UBL, *unblocking message*) y se presenta a la entidad de protocolo de gestión recién creada. Antes de la devolución, se asigna y rellena un nuevo registro del tipo "MRec".

El procedimiento "**ResetConn**" (de la parte 18 de la figura B.3) llama a la función "FindNextCRec" para encontrar los enlaces AAL tipo 2 afectados por la operación de reiniciación. Para todos estos enlaces se terminan las entidades de protocolo asociadas. En el punto de servicio AAL tipo 2 extremo, se envía una primitiva RELEASE.confirm al usuario servido. En el conmutador AAL tipo 2, se inician los procedimientos de liberación para el usuario remoto servido. Se comprueba la liberación de los recursos asignados a estos enlaces; los recursos de los trayectos AAL tipo 2 que se reinician son liberados por el propio procedimiento de reiniciación.

El procedimiento "**ResetBlock**" (de la parte 18 de la figura B.3) garantiza que para todos los trayectos AAL tipo 2 "localmente bloqueados" afectados por una operación de reiniciación (iniciada por otro nodo) se inicie un procedimiento de bloqueo (antes de que se confirme la reiniciación).

El procedimiento **"ResetUnblock"** (de la parte 18 de la figura B.3) poner todos los trayectos AAL tipo 2 "bloqueados a distancia" afectados por la operación de reiniciación, a "desbloqueados a distancia".

#### **B.4.2.4 Macros**

Las macros se describen en las partes 22 a 25 (de 25) de la figura B.3.

La macro **"Construct ERQmsg"** (de la parte 22 de la figura B.3) proporciona los detalles necesarios para la construcción del mensaje ERQ. En particular, se añaden parámetros al mensaje dependiendo de los parámetros de la primitiva ESTABLISH.request procedente del usuario servido.

La macro **"ReturnConfuse"** (de la parte 23 de la figura B.3) devuelve el mensaje confusión (CNF) al remitente del último mensaje que provocó la confusión.

La macro **"ReturnReject"** (de la parte 23 de la figura B.3) construye el mensaje de confirmación de liberación (RLC) con el parámetro causa (CAU) y lo devuelve al remitente del mensaje de petición establecimiento (ERQ); la macro **"SendReject"** construye el mensaje de confirmación de liberación (RLC) con el parámetro causa (CAU, *cause parameter*) y lo envía al remitente del mensaje de petición de establecimiento (ERQ) una vez procesado el mensaje ERQ, por ejemplo, si se ha reiniciado el enlace siguiente.

La macro **"ReturnRelease"** (de la parte 23 de la figura B.3) construye el mensaje liberación (REL) y lo devuelve al remitente del último mensaje; esto puede suceder como respuesta a un mensaje no reconocido o que provocara confusión. La macro **"SendRelease"** construye el mensaje liberación (REL) y lo envía al enlace precedente o al siguiente. Esto puede ocurrir como respuesta a un mensaje no reconocido o con la operación reiniciación.

La macro **"Extract ERQparameters"** (de la parte 24 de la figura B.3) extrae la información del mensaje de petición de establecimiento (ERQ) y prepara los parámetros de la primitiva ESTABLISH.indication.

La macro **"ValidREL"** (de la parte 24 de la figura B.3) garantiza la existencia del parámetro causa (CAU) en el mensaje liberación (REL); si falta este parámetro, se añade dentro de esta macro ("normal, no especificado", sin diagnósticos). La macro **"ValidRLC"** verifica si hay un parámetro causa (CAU) en el mensaje liberación completa (RLC) cuando no se espera ninguno; si el parámetro está presente la causa (que resulta de una confusión) se comunica a la gestión de capa. La macro **"ValidRLCR"** garantiza la existencia del parámetro causa (CAU) en el mensaje liberación completa (RLC) cuando se espera; si falta este parámetro se añade dentro de esta macro ("fallo temporal", sin diagnósticos). La macro **"ValidCNF"** verifica si hay un parámetro causa (CAU) en el mensaje confusión (CNF); no se añade parámetro en este caso.

Las macros **"Construct ECFmsg"**, **"Construct RLCmsg"**, **"Construct RSCmsg"**, **"Construct BLCmsg"** y **"Construct UBCmsg"** (de la parte 25 de la figura B.3) indican que los respectivos mensajes se construyen sin parámetros.

Las macros **"Construct RESmsg"**, **"Construct BLOmsg"**, y **"Construct UBLmsg"** (de la parte 25 de la figura B.3) indican que los mensajes respectivos se construyen con el parámetro identificador del elemento de conexión (CEID, *connection element identifier parameter*).

Las macros **"Construct RELmsg"**, **"Construct RLCmsgR"**, **"Construct RSCmsgC"**, **"Construct BLCmsgC"**, **"Construct UBCmsgC"**, y **"Construct RELmsg"** y **"Construct CNFmsg"** (de la parte 25 de la figura B.3) indican que los respectivos mensajes se construyen con el parámetro causa (CAU).

NOTA – Los mensajes liberación (REL) y confusión (CNF) siempre contienen el parámetro causa. El mensaje confirmación de liberación (RLC) contiene el parámetro causa para responder al mensaje de petición de establecimiento (ERQ) (rechazo del establecimiento) o para responder al mensaje liberación acompañado de notificaciones de compatibilidad. Los mensajes confirmación de reiniciación (RSC), confirmación de bloqueo (BLC) y confirmación de desbloqueo (UBC) contienen el parámetro causa cuando se acompañan de notificaciones de compatibilidad.

### B.4.3 Procedimientos en la función nodal

La función "**PathRes**" lleva a cabo el control de admisión de conexión seguido por la reserva de recursos en un enlace entrante (el enlace precedente) durante el establecimiento de la conexión y devuelve el valor definido por la siguiente estructura ASN.1:

```

PathRes ::= SEQUENCE {
    course          ENUMERATED {success, fail},
    cause          CAUSE }
-- return value of procedure
-- PathResource
-- cause if "course = fail"

```

NOTA – No se detallan en esta Recomendación el control de admisión de la conexión ni la reserva de los recursos.

La función "**SelectRoute**" adopta la decisión de encaminamiento seguida de la reserva de recursos en el enlace saliente (el enlace siguiente) durante el establecimiento de la conexión.

La función nodal determina la disponibilidad de una ruta con suficientes recursos de trayecto AAL tipo 2 para el nodo AAL tipo 2 subsiguiente. Esto puede incluir una función para seleccionar una ruta con recursos disponibles para el nodo AAL tipo 2 siguiente. A continuación selecciona un trayecto AAL tipo 2 del interior de dicha ruta, capaz de acomodar la nueva conexión.

El encaminamiento suele realizarse en base a:

- la información de direccionamiento (en el caso conmutado);
- el indicador de conexión de prueba;
- la información del enlace (características del enlace); y
- otras informaciones (incluida la información SSCS).

Esta función devuelve el valor definido por la siguiente estructura ASN.1:

```

Route ::= SEQUENCE {
    course          ENUMERATED {remote, local, fail},
    ceid           CEID,
    sri            PID,
    cause          CAUSE }
-- return value of procedure SelectRoute
-- connection element identifier
-- nodal signalling association
-- identifier
-- cause if "fail"

```

La función "**SwitchRoute**" (de la función nodal) adopta la decisión de encaminamiento seguida de la reserva de recursos dentro del nodo AAL tipo 2.

Esta ruta se establece entre el usuario servido AAL tipo 2 solicitante o el enlace entrante (precedente) por una parte y el usuario servido AAL tipo 2 de destino o el enlace saliente (subsiguiente) por la otra durante el establecimiento de la conexión. Se devuelve el valor definido por la siguiente estructura ASN.1:

```

Switch ::= SEQUENCE {
    course          ENUMERATED {success, fail},
    cause          CAUSE }
-- return value of procedure
-- SwitchRoute
-- cause if "fail"

```

El procedimiento "**PathRel**" libera los recursos asociados con el trayecto AAL tipo 2 que se designan por "CRec.incoming" o "CRec.outgoing".

El procedimiento "**SwitchRel**" libera los recursos asociados con la conexión AAL tipo 2 dentro del nodo AAL tipo 2.

El procedimiento "**ResetRel**" libera los recursos de la conexión AAL tipo 2 asociados con (uno de) los canales sometidos al procedimiento de mantenimiento de reiniciación.

El procedimiento "**AddCompatibility**" completa un mensaje con la información de compatibilidad apropiada en el campo de compatibilidad del mensaje así como en todos los campos de compatibilidad de los parámetros.

La función "**PassOnPossible**" devuelve el valor "TRUE" si es posible el paso del mensaje no reconocido o de un parámetro no reconocido dentro de un mensaje reconocido; de lo contrario, se devuelve "FALSE".

La función "**ParamKnown**" devuelve el valor "TRUE" si el parámetro se reconoce; de lo contrario se devuelve "FALSE".

La función "**FieldRecognized**" devuelve el valor "TRUE" si se reconoce el valor del campo; de lo contrario, se devuelve "FALSE".

La función "**PassConfuse**" devuelve el valor "TRUE" si se debe pasar el mensaje CNF recibido; de lo contrario, se devuelve "FALSE".

La función "**Valid**" devuelve el valor "TRUE" si es válido el CEID dentro de una relación de señalización dada; de lo contrario, se devuelve "FALSE".

La función "**ValidateLink**" determina si la información CEID en ERQmsg designa un enlace válido en la asociación de señalización nodal "sri". Si la información es válida devuelve el valor "TRUE" (de lo contrario devuelve "FALSE").

El procedimiento "**BLOCKING**" pone el estado de un trayecto (dentro de una relación de señalización) a bloqueado o desbloqueado localmente y bloqueado o desbloqueado a distancia.

La función "**Blocked**" devuelve el valor "TRUE" si el trayecto AAL tipo 2 indicado está bloqueado localmente o a distancia (como lo indica el segundo parámetro); de lo contrario, se devuelve "FALSE".

#### **B.4.4 Estructura de datos de los mensajes y parámetros de señalización AAL tipo 2**

Los diagramas SDL utilizan la siguiente estructura y definición ASN.1 de los mensajes y parámetros de señalización AAL tipo 2:

##### **B.4.4.1 Mensaje genérico y estructura de los parámetros**

```
Message ::= SEQUENCE {
    dsaid          SAID,
    msgID          MessageID,
    msgcompat      BIT STRING (SIZE (8)),
    parameters     SET OF Parameter }
-- general definition of the AAL type 2 signalling
-- destination signalling association ID
-- message identifier
-- message compatibility
-- message parameters
messages
```

```

Parameter ::= SEQUENCE { -- general definition of the AAL type 2 signalling
                        parameters
    paramid             BIT STRING (SIZE (8)), -- parameter identifier
    paramcompat        BIT STRING (SIZE (8)), -- parameter
                                                compatibility
    paramlength        INTEGER (0 .. 255),    -- parameter length
    fields              SEQUENCE OF Fields }  -- parameter fields

```

```

Fields ::= CHOICE { -- general definition of the AAL type 2
                   signalling parameters
    fldtyp1          FixedSizeField,
    fldtyp2          VariableSizeField }
FixedSizeField ::= fixedfield OCTET STRING (SIZE (1 .. 255))
VariableSizeField ::= SEQUENCE {
    fieldlength      INTEGER (0 .. 254),      -- field
                                                length
    variablefield    OCTET STRING (SIZE (0 .. 254)) }

```

MessageID	BIT STRING (SIZE (8))	-- message identifier
erq	MessageID ::= '00000101'H	-- Establish request
ecf	MessageID ::= '00000100'H	-- Establish confirm
rel	MessageID ::= '00000111'H	-- Release request
rlc	MessageID ::= '00000110'H	-- Release confirm
res	MessageID ::= '00001001'H	-- Reset request
rsc	MessageID ::= '00001000'H	-- Reset confirm
blo	MessageID ::= '00000010'H	-- Block request
blc	MessageID ::= '00000001'H	-- Block confirm
ubl	MessageID ::= '00001011'H	-- Unblock request
ubc	MessageID ::= '00001010'H	-- Unblock confirm
cnf	MessageID ::= '00000011'H	-- Confusion

#### B.4.4.2 Estructura detallada de los parámetros

```

CAU ::= SEQUENCE { -- definition of the essentials of the cause
                  (CAU) parameter
    org             BIT STRING (SIZE (2)) ('00'B), -- origin
                                                            ITU-T
    value           INTEGER (1 .. 127),          -- cause
                                                            value
    diagnostics     OCTET STRING (SIZE (0 .. 252)) }

```

```

CEID ::= SEQUENCE { -- definition of the essentials of the
                   CEID parameter
    path            AAL2Path,                    -- AAL type
                                                            2 path
    cid             CID }

```

```

OSAID ::= SAID -- definition of the OSAID
           parameter
SAID ::= OCTET STRING (SIZE (4)) -- definition of
           the SAID
unknown SAID ::= '00000000'H -- definition of
           the 'unknown'
           value

```

```

TCI ::= OCTET STRING (SIZE (0)) -- definition of the essentials of the TCI
           parameter

```

```

-- The following parameters are handled but never interpreted in the
-- SDL definition, hence, no details are needed
ESEA ::= OCTET STRING (SIZE (3 .. 17)) -- definition of the essentials of
           the ESEA parameter
NSEA ::= OCTET STRING (SIZE (20)) -- definition of the essentials of
           the NSEA parameter
ALC ::= OCTET STRING (SIZE (12)) -- definition of the essentials of
           the ALC parameter
SUGR ::= OCTET STRING (SIZE (4)) -- definition of the essentials of
           the SUT parameter
SUT ::= OCTET STRING (SIZE (1 .. 255)) -- definition of the essentials of
           the SUGR parameter
SSIA ::= OCTET STRING (SIZE (8)) -- definition of the essentials of
           the SSIA parameter
SSIM ::= OCTET STRING (SIZE (3)) -- definition of the essentials of
           the SSIM parameter
SSISA ::= OCTET STRING (SIZE (14)) -- definition of the essentials of
           the SSISA parameter
SSISU ::= OCTET STRING (SIZE (7)) -- definition of the essentials of
           the SSISU parameter

```

NOTA – Los parámetros no interpretados por la función nodal 1 se representan sólo en forma resumida; sin embargo, estos parámetros con la excepción de "SUGR" y "SUT" son interpretados por la función nodal 2.

#### B.4.4.3 Relación de la estructura de los parámetros detallados para los mensajes

```

ERQmsg ::= SEQUENCE { -- definition of the essentials of the ERQ message
  ceid      CEID, -- connection element identifier
  a2ea     CHOICE { -- AAL type 2 endpoint address
    esea     ESEA, -- destination E.164 endpoint
              address
    nsea     NSEA }, -- destination NSAP endpoint
              address
  alc      ALC OPTIONAL, -- link characteristics
  osaid    OSAID, -- originating signalling
              association ID
  sugr     SUGR OPTIONAL, -- served user generated reference
  sut      SUT OPTIONAL, -- serverd user transport
  ssis     CHOICE { -- SSCS information
    ssia     SSIA, -- service specific information
              (audio)
    ssim     SSIA, -- service specific info.
              (multirate)
    ssisa    SSIA, -- service specific info.
              (SAR-assured)
    ssisu    SSIA } OPTIONAL, -- service specific info.
              (SAR-unassured)
  tci      TCI OPTIONAL } -- test connection indicator

```

```
ECFmsg ::= SEQUENCE { -- definition of the essentials of the ECF message
    osaid          OSAID } -- originating signalling
                                association ID
```

```
RELmsg ::= SEQUENCE { -- definition of the essentials of the REL message
    cause          CAU } -- cause
```

```
RLCmsg ::= SEQUENCE { -- definition of the essentials of the RLC message
    cause          CAU OPTIONAL } -- cause
```

```
RESmsg ::= SEQUENCE { -- definition of the essentials of the RES message
    ceid           CEID, -- connection element identifier
    osaid          OSAID } -- originating signalling
                                association ID
```

```
RSCmsg ::= SEQUENCE { -- definition of the essentials of the RSC message
    cause          CAU OPTIONAL } -- cause
```

```
BLOmsg ::= SEQUENCE { -- definition of the essentials of the BLO message
    ceid           CEID, -- connection element identifier
    osaid          OSAID } -- originating signalling
                                association ID
```

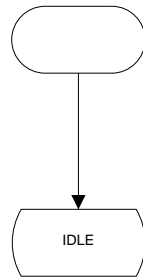
```
BLCmsg ::= SEQUENCE { -- definition of the essentials of the BLC message
    cause          CAU OPTIONAL } -- cause
```

```
UBLmsg ::= SEQUENCE { -- definition of the essentials of the UBL message
    ceid           CEID, -- connection element identifier
    osaid          OSAID } -- originating signalling
                                association ID
```

```
UBCmsg ::= SEQUENCE { -- definition of the essentials of the UBC message
    cause          CAU OPTIONAL } -- cause
```

```
CNFmsg ::= SEQUENCE { -- definition of the essentials of the CNF message
    cause          CAU } -- cause
```





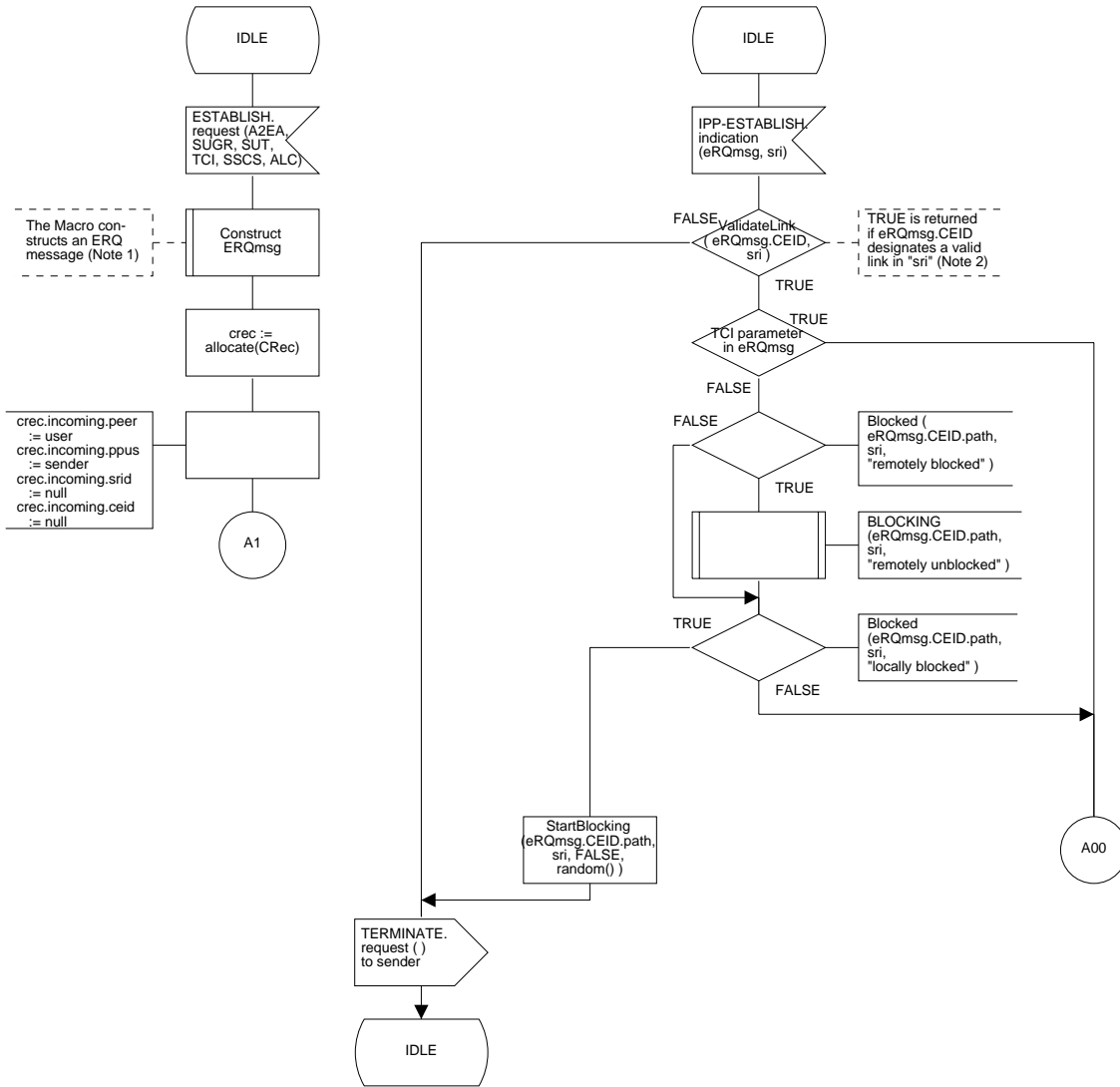
```

dcl
eRQmsg ERQmsg, /* Establish request */
eCFmsg ECFmsg, /* Establish confirm */
rELmsg RELmsg, /* Release request */
rLCmsg RLCmsg, /* Release confirm */
rESmsg RESmsg, /* Reset request */
rSCmsg RSCmsg, /* Reset confirm */
bLOmsg BLOmsg, /* Block request */
bLCmsg BLCmsg, /* Block confirm */
uBLmsg UBLmsg, /* Unblock request */
uBCmsg UBCmsg, /* Unblock confirm */
cNFmsg CNFmsg: /* Confusion */
  
```

## NOTE

At start-up, it is assumed that the STI (Signalling Transport Interfaces) to each existing STC (Signalling Transport Converter, there exists one per nodal signalling relation) are created. The addition or removal of nodal signalling relations together with the creation or destruction of the STI and STC processes is not shown in the SDL diagrams of this Annex.

**Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 1 de 25)**



**NOTES**

1. The ERQ message includes the required parameters but without compatibility information filled in.
2. The procedure "ValidateLink" determines whether the CEID information in the ERQmsg designates a valid link in the nodal signalling association "sri"; it is located with the procedures in NodalF2 and not further specified.
3. The procedures and functions "Blocked" and "BLOCKING", are located in NodalF2. "StartBlocking" is located in NodalF1.

Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 2 de 25)



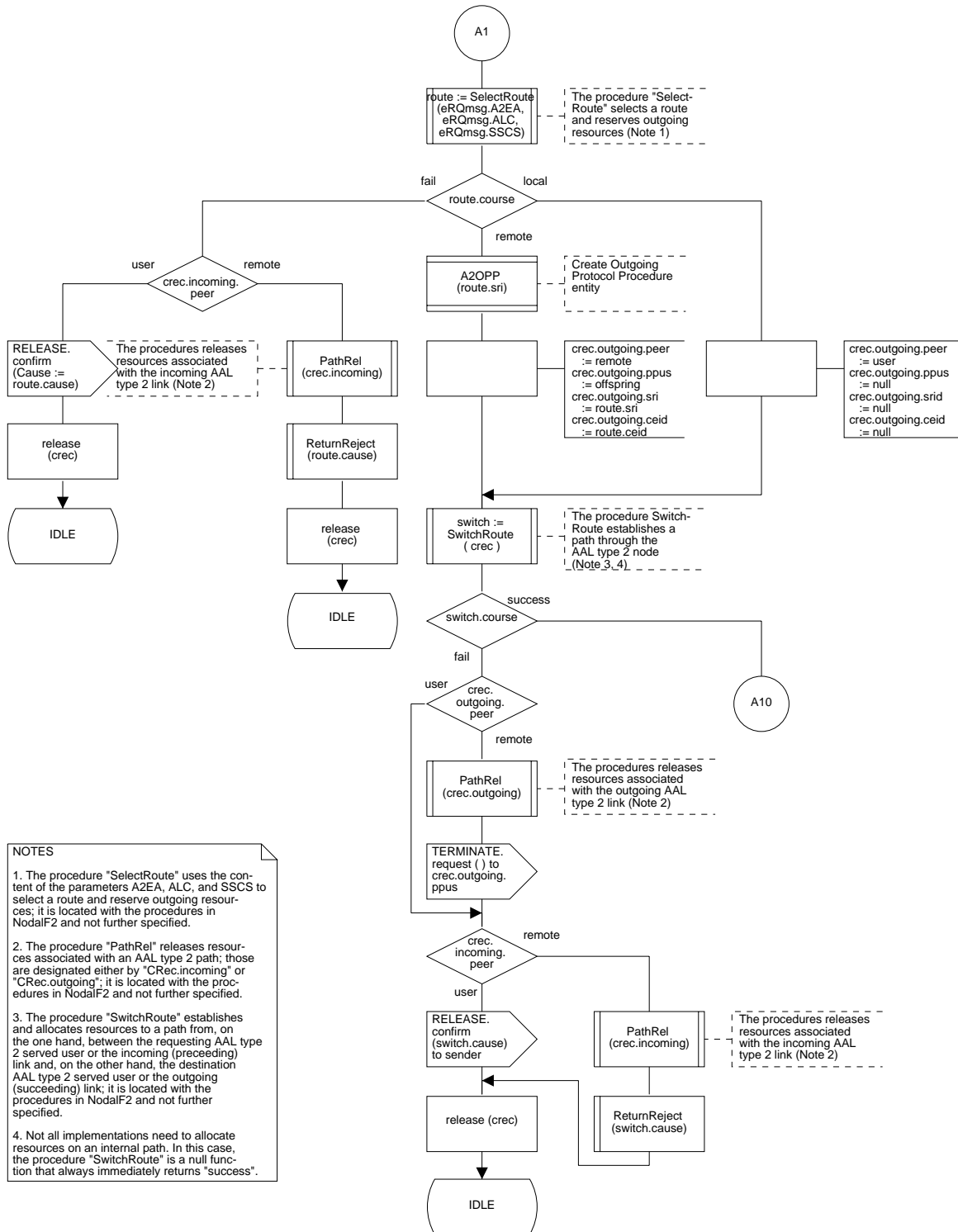


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 4 de 25)

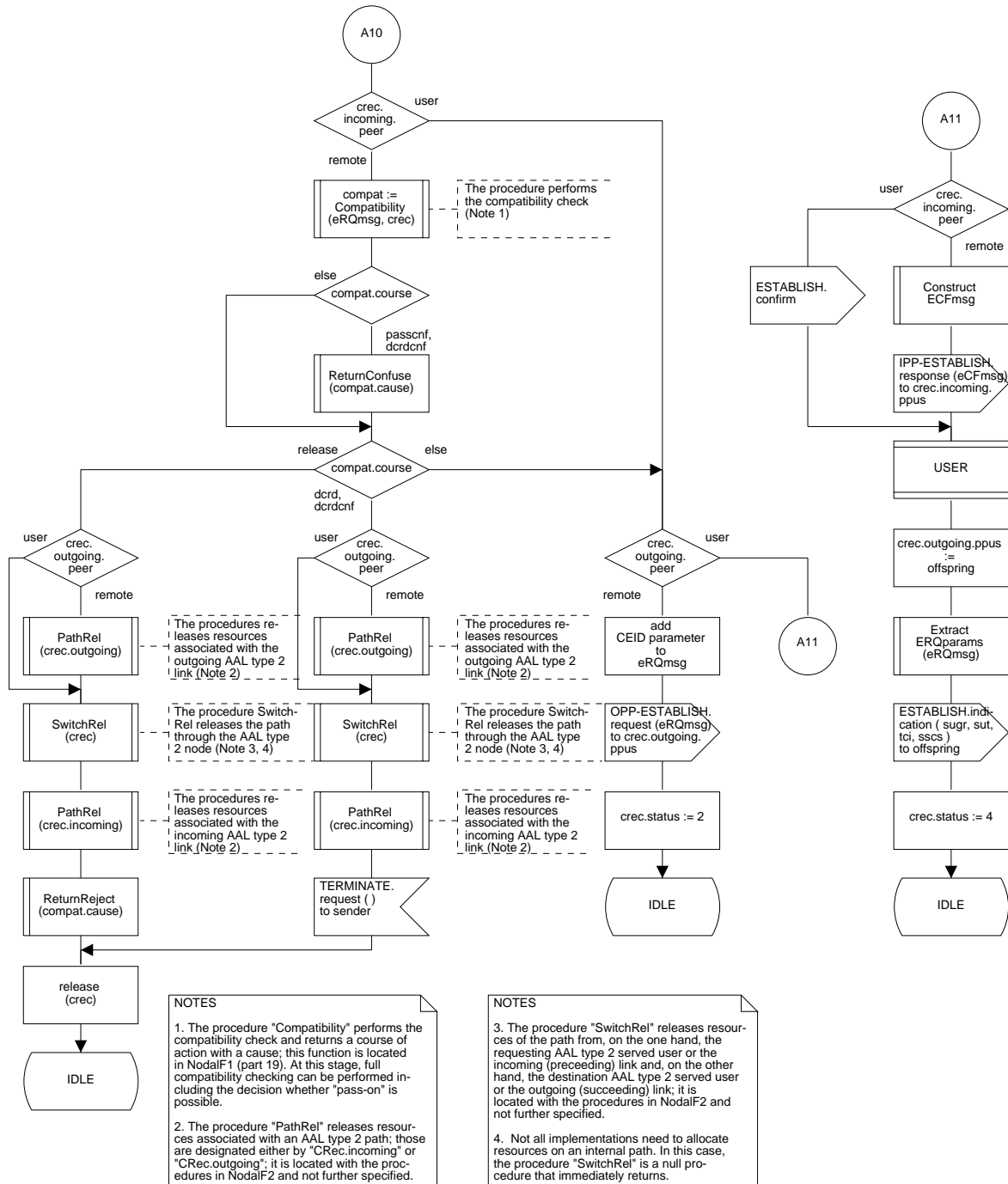


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 5 de 25)

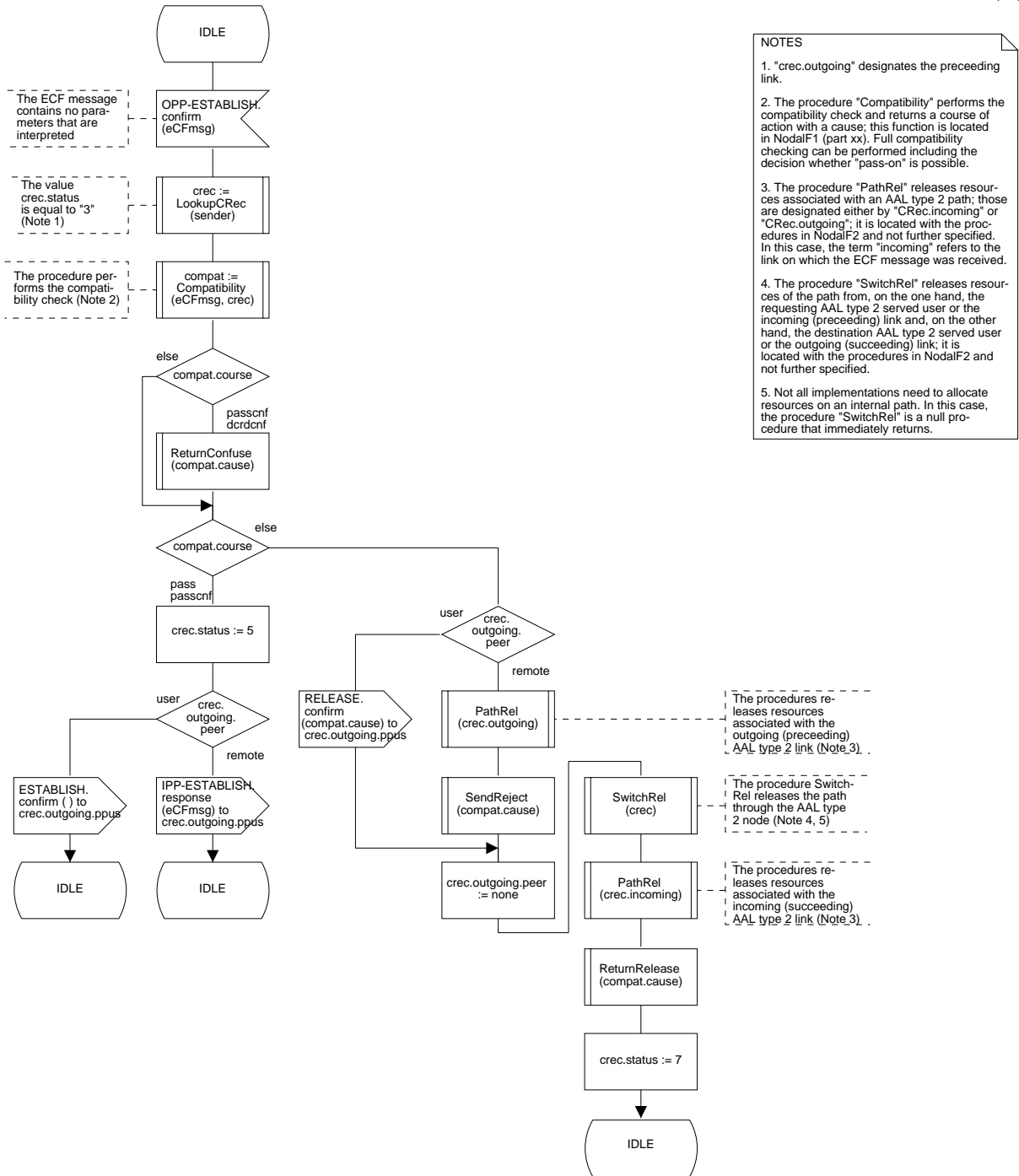
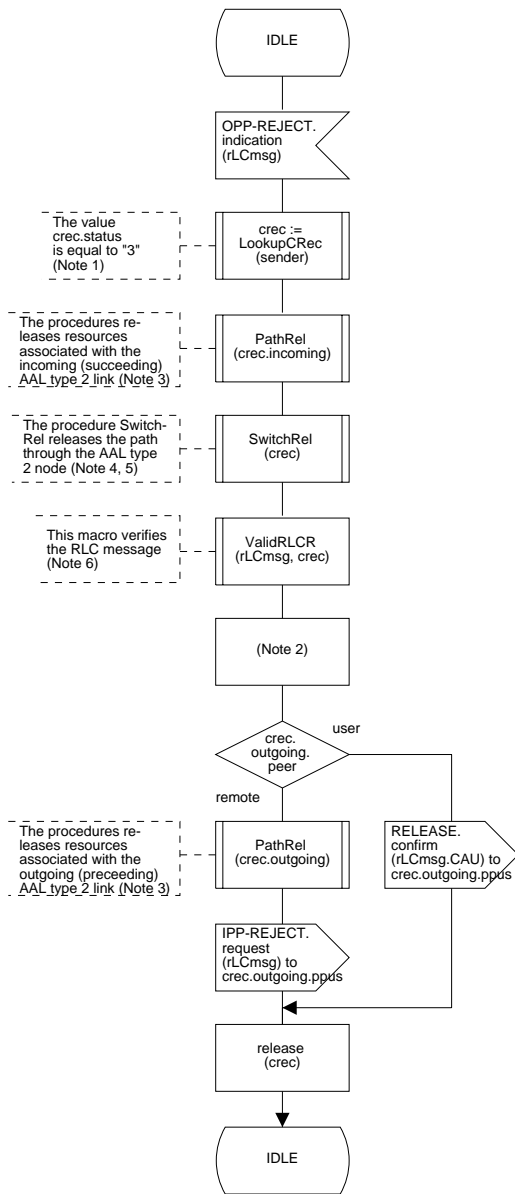
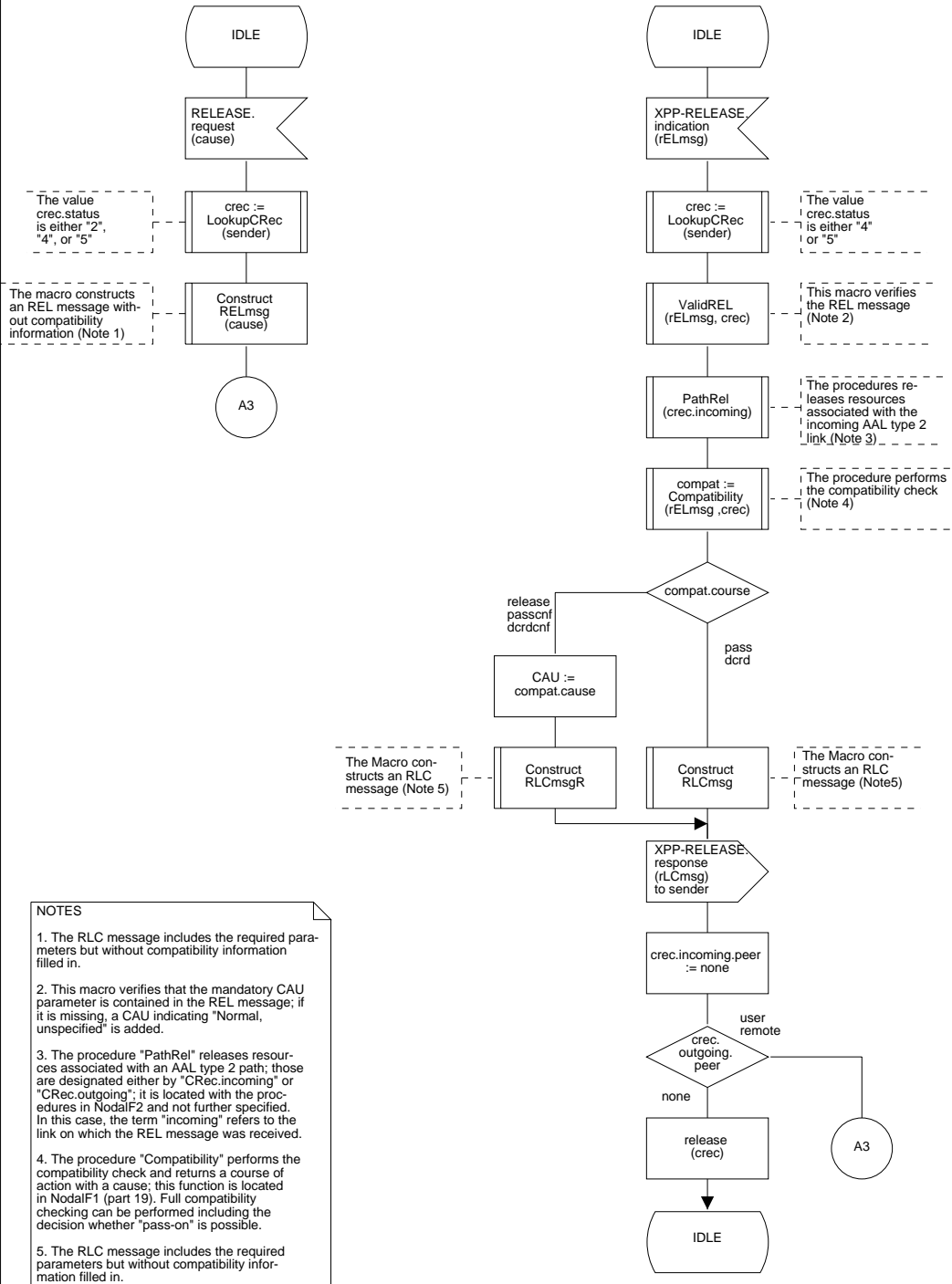


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 6 de 25)



- NOTES**
1. "crec.outgoing" designates the preceeding link.
  2. Features that enable a further connection attempt, involving the selection of a different AAL type 2 path within the same route or of an alternative route, may be implemented; however, such features are not specified.
  3. The procedure "PathRel" releases resources associated with an AAL type 2 path; those are designated either by "CRec.incoming" or "CRec.outgoing"; it is located with the procedures in NodalF2 and not further specified. In this case, the term "incoming" refers to the link on which the ECF message was received.
  4. The procedure "SwitchRel" releases resources of the path from, on the one hand, the requesting AAL type 2 served user or the incoming (preceeding) link and, on the other hand, the destination AAL type 2 served user or the outgoing (succeeding) link; it is located with the procedures in NodalF2 and not further specified.
  5. Not all implementations need to allocate resources on an internal path. In this case, the procedure "SwitchRel" is a null procedure that immediately returns.
  6. This macro verifies that the mandatory CAU parameter is contained in the RLC message; if it is missing, a CAU indicating "Temporary failure" is added.

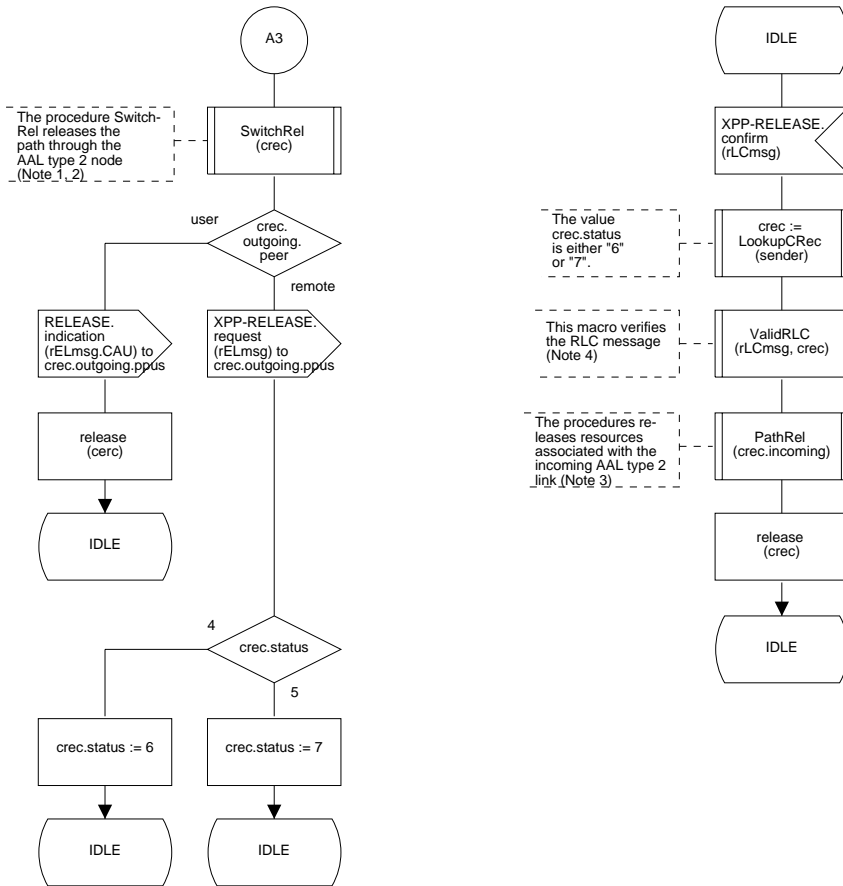
Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 7 de 25)



- NOTES**
1. The RLC message includes the required parameters but without compatibility information filled in.
  2. This macro verifies that the mandatory CAU parameter is contained in the REL message; if it is missing, a CAU indicating "Normal, unspecified" is added.
  3. The procedure "PathRel" releases resources associated with an AAL type 2 path; those are designated either by "CRec.incoming" or "CRec.outgoing"; it is located with the procedures in NodalF2 and not further specified. In this case, the term "incoming" refers to the link on which the REL message was received.
  4. The procedure "Compatibility" performs the compatibility check and returns a course of action with a cause; this function is located in NodalF1 (part 19). Full compatibility checking can be performed including the decision whether "pass-on" is possible.
  5. The RLC message includes the required parameters but without compatibility information filled in.

Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 8 de 25)





- NOTES**
1. The procedure "SwitchRel" releases resources of the path from, on the one hand, the requesting AAL type 2 served user or the incoming (preceeding) link and, on the other hand, the destination AAL type 2 served user or the outgoing (succeeding) link; it is located with the procedures in NodalF2 and not further specified.
  2. Not all implementations need to allocate resources on an internal path. In this case, the procedure "SwitchRel" is a null procedure that immediately returns.
  3. The procedure "PathRel" releases resources associated with an AAL type 2 path; those are designated either by "CRec.incoming" or "CRec.outgoing"; it is located with the procedures in NodalF2 and not further specified. In this case, the term "incoming" refers to the link on which the RLC message was received.
  4. This macro notifies layer management if the optional CAU parameter is contained in the RLC message.

Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 9 de 25)

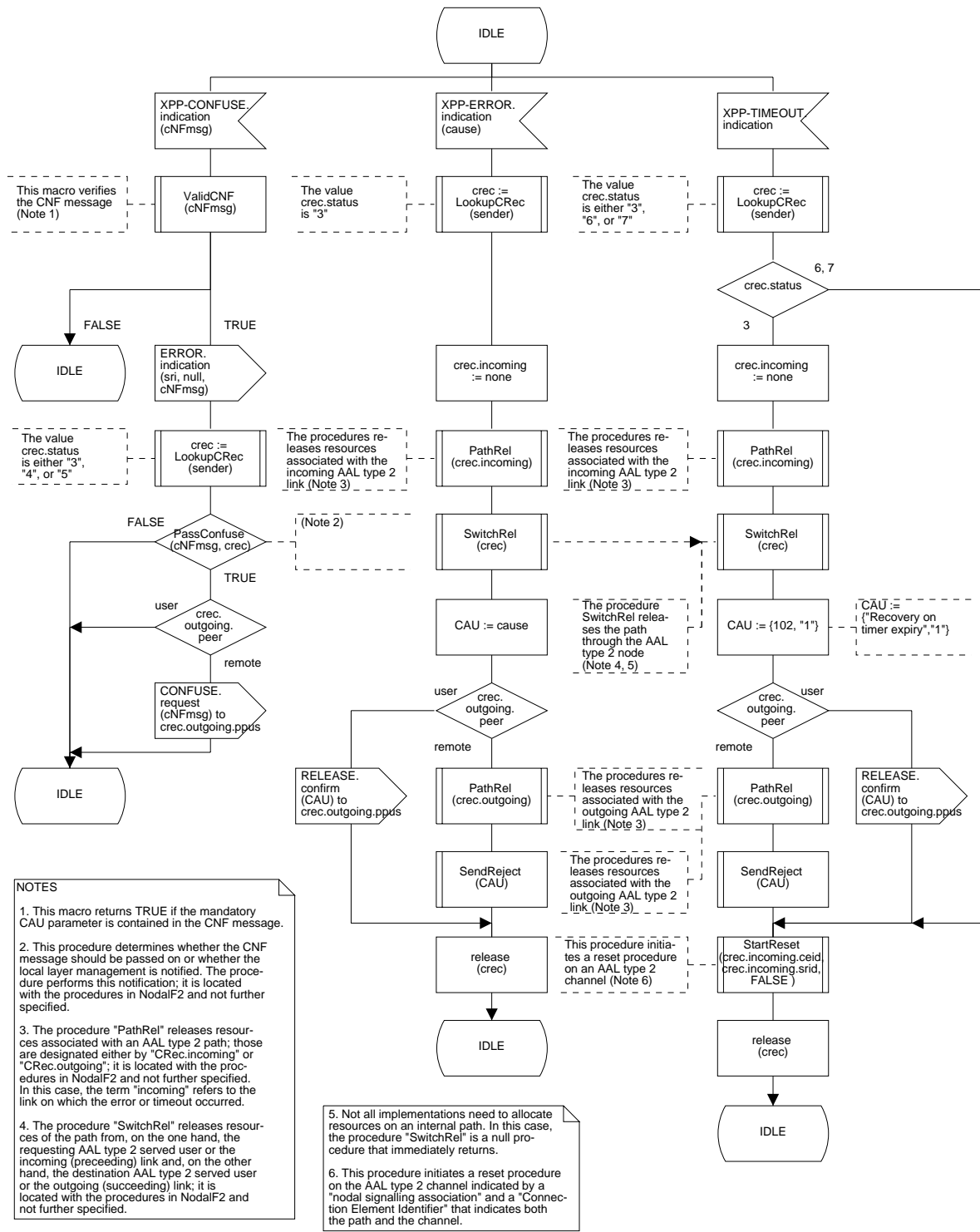


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 10 de 25)

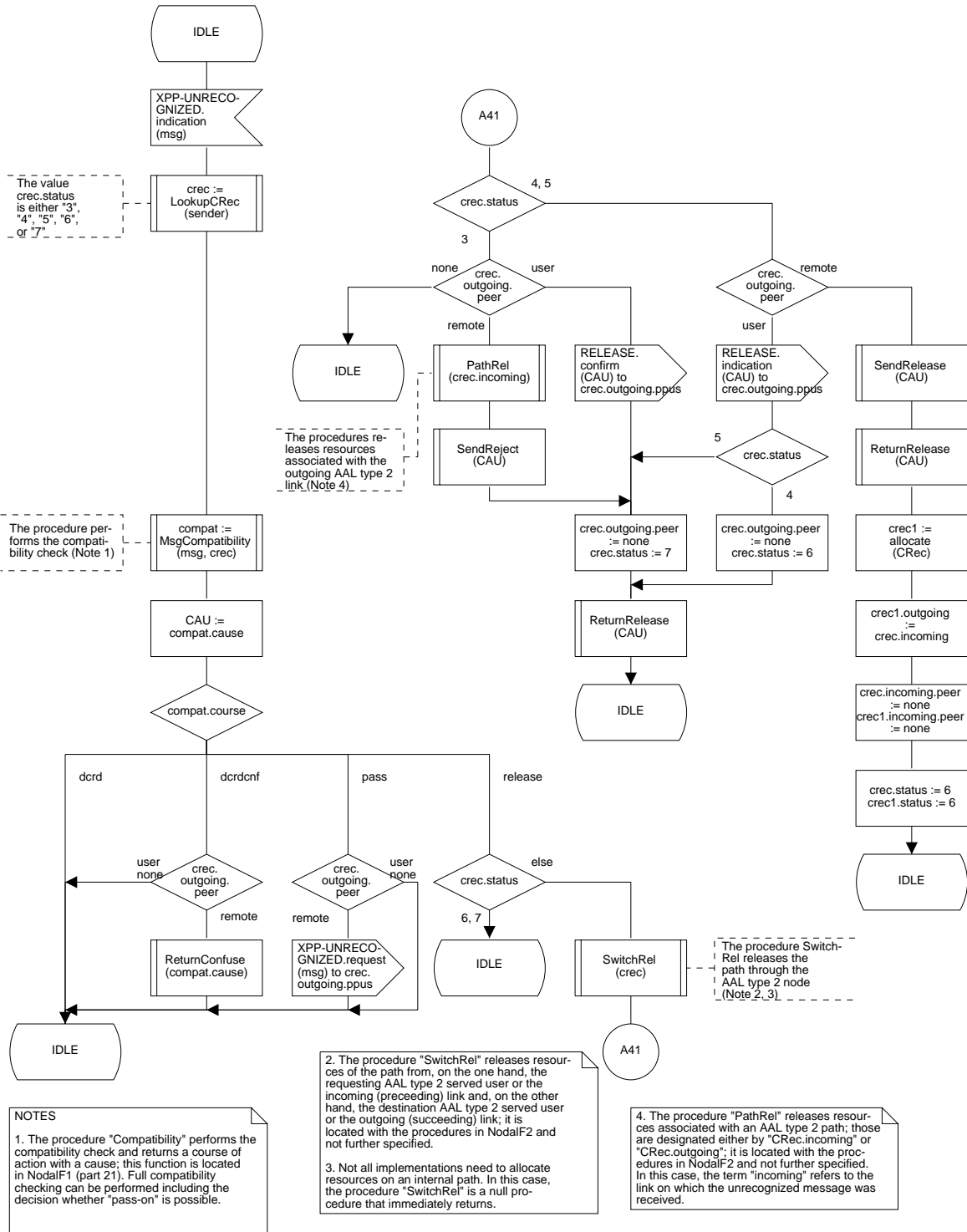


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 11 de 25)

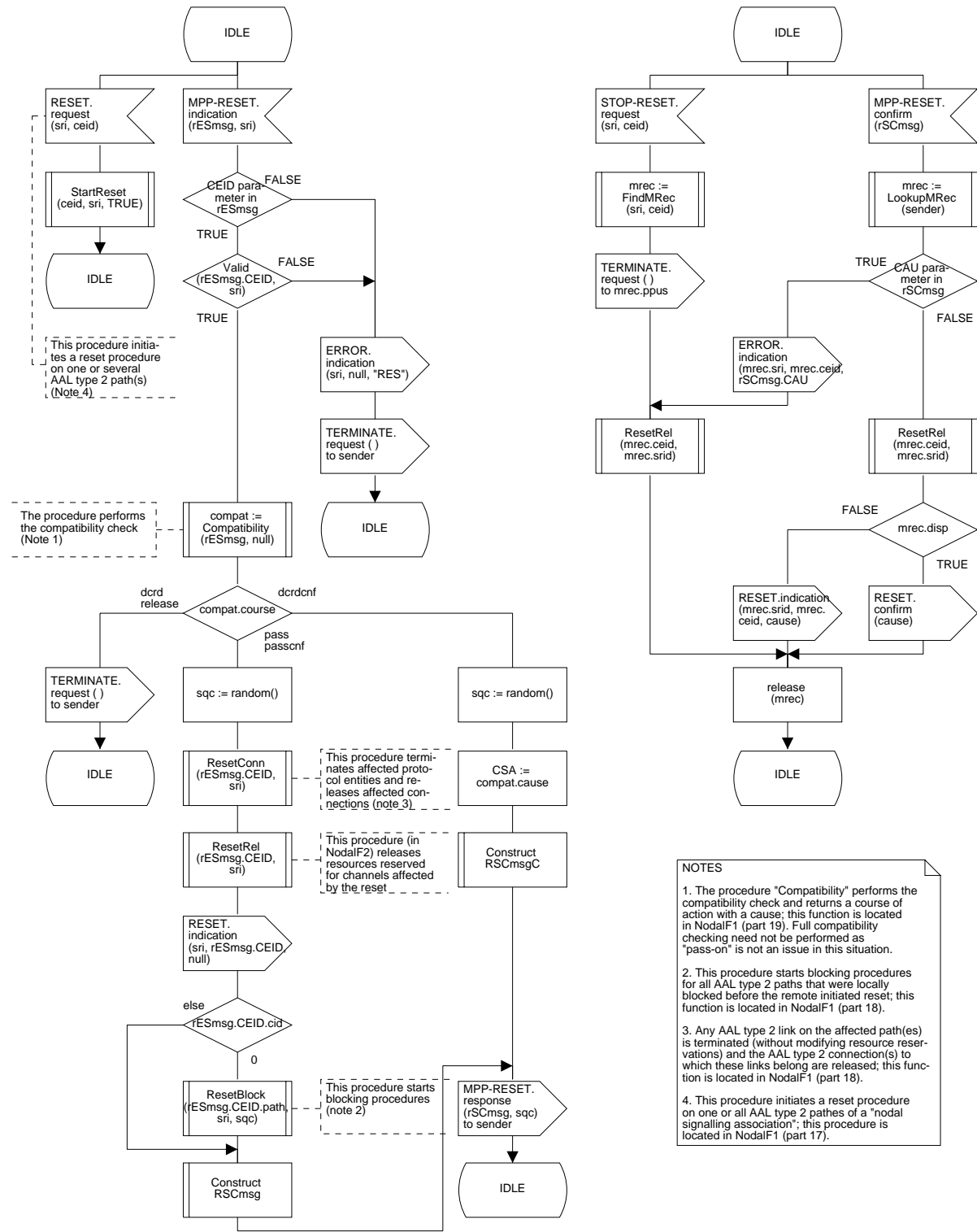
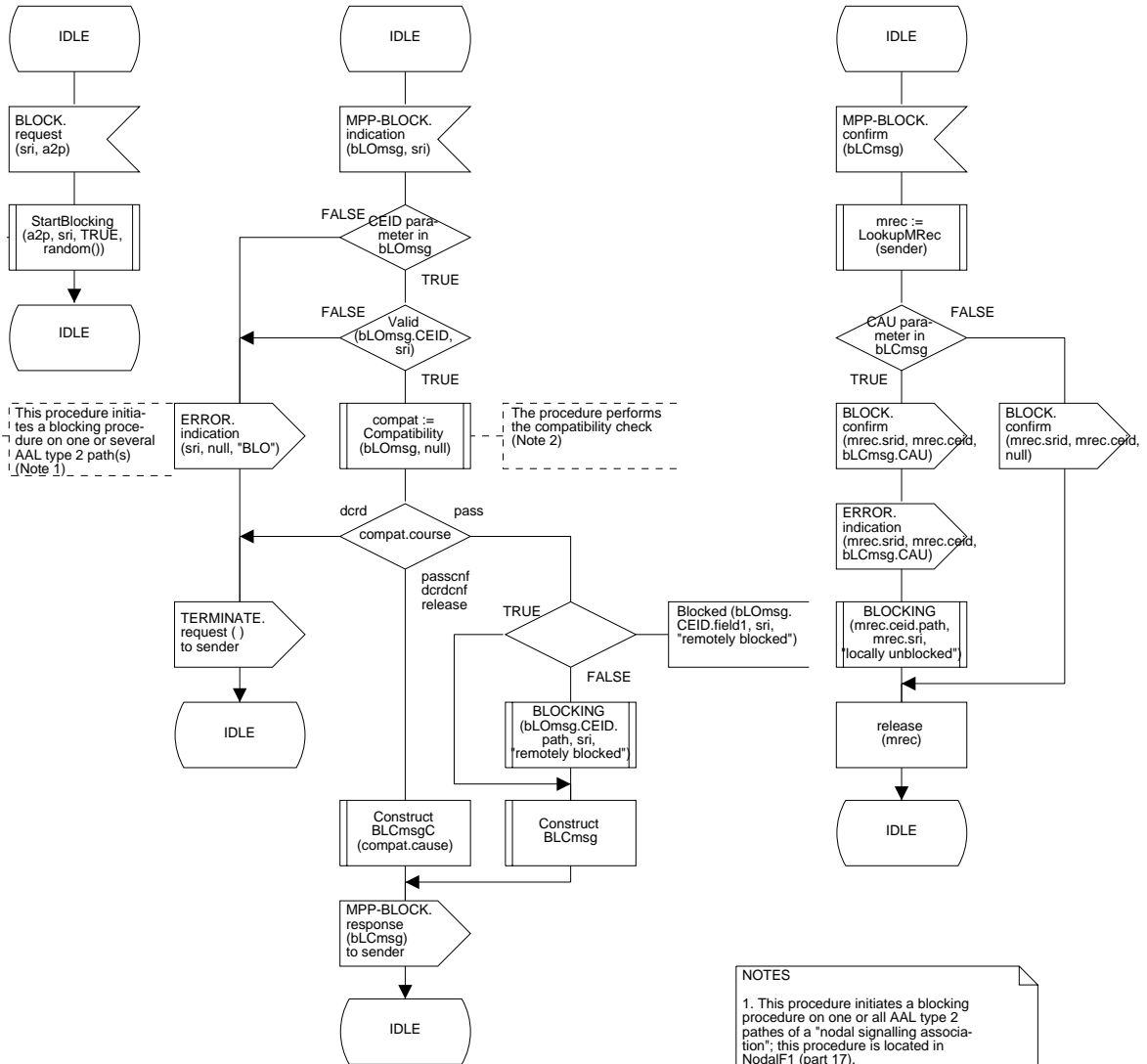


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 12 de 25)



**NOTES**

1. This procedure initiates a blocking procedure on one or all AAL type 2 paths of a "nodal signalling association"; this procedure is located in NodalF1 (part 17).
2. The procedure "Compatibility" performs the compatibility check and returns a course of action with a cause; this function is located in NodalF1 (part 19). Full compatibility checking need not be performed as "pass-on" is not an issue in this situation.

Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 13 de 25)

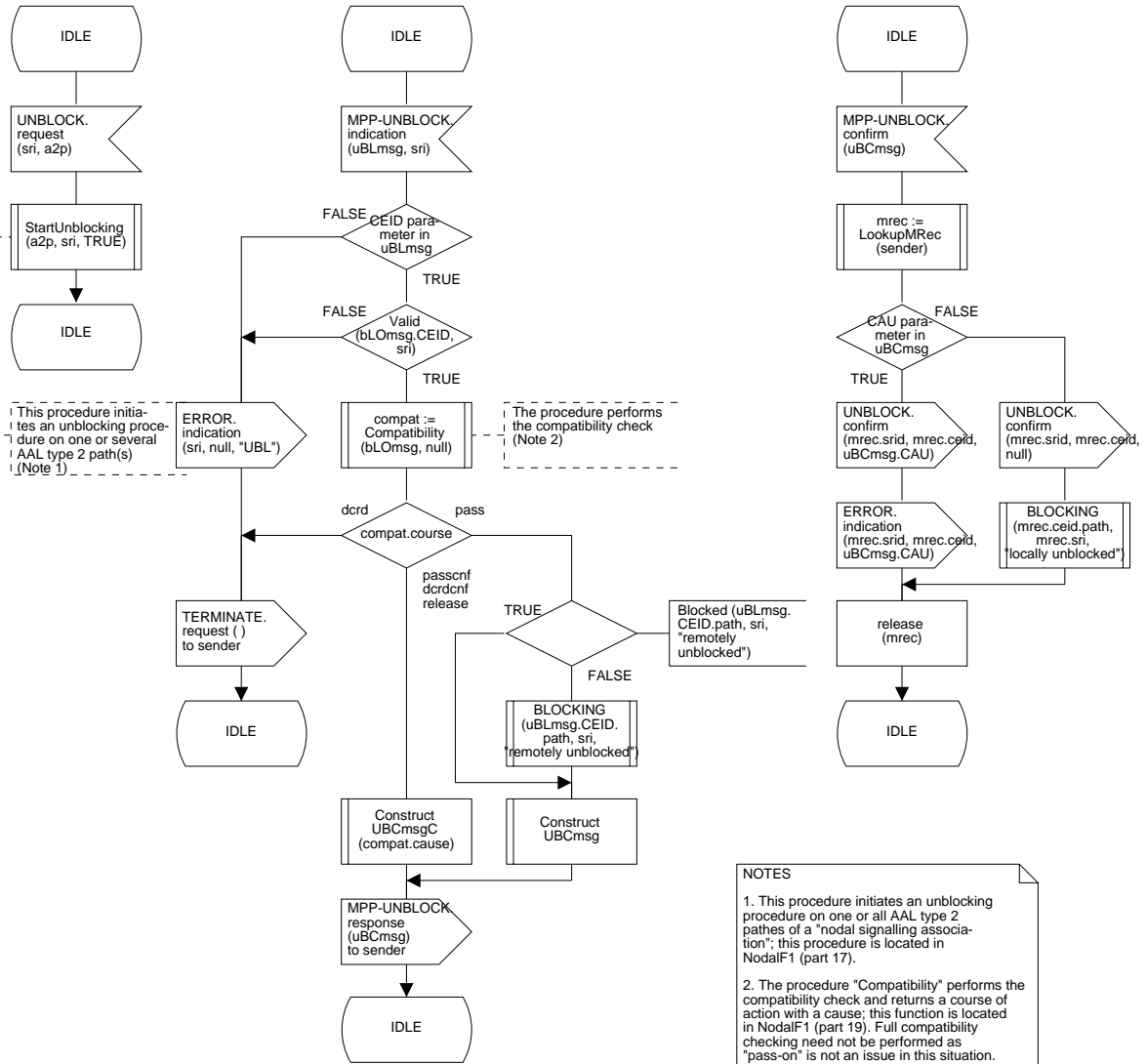


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 14 de 25)

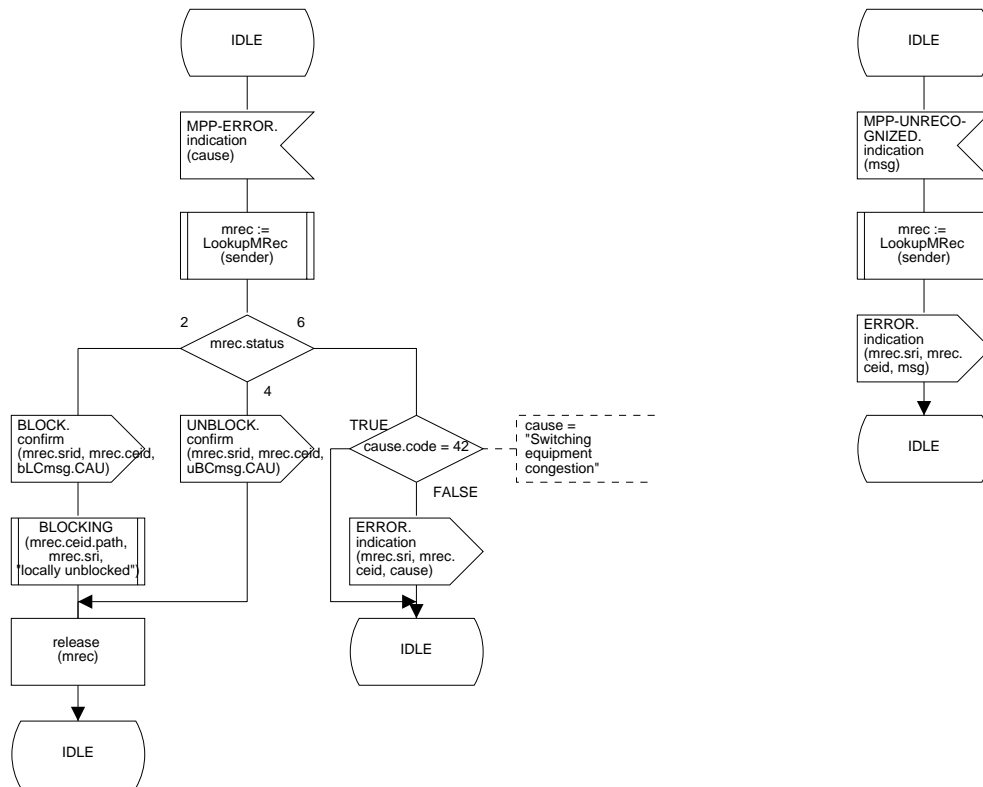
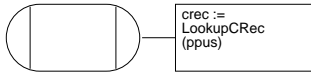


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 15 de 25)



This function searches all records of type "CRec" to find the one that matches either the crec.incoming.ppus or the crec.outgoing.ppus with the input parameter. Exactly one such record is found.

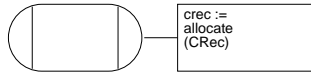
If the input parameter matches the crec.outgoing.ppus, the incoming and outgoing parts of the record are exchanged. If such an exchange took place, the status part of the record is also modified as follows:

```

if even(crec.status) then
  increment crec.status by 1
else
  decrement crec.status by 1
endif

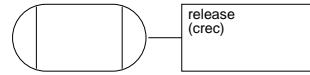
```

The value returned can be understood to be a pointer to the record itself.

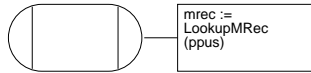


This function allocates a record of type "CRec" or "MRec".

The value returned can be understood to be a pointer to the record itself.

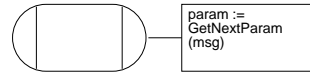


This function deallocates a record of type "CRec" or "MRec" referenced by the parameter "crec" or "mrec". The record becomes unavailable.



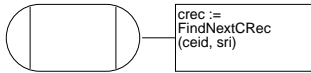
This function searches all records of type "MRec" to find the one that matches the mrec.ppus parameter. Exactly one such record is found.

The value returned can be understood to be a pointer to the record itself.



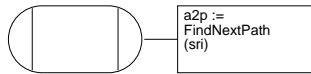
This function parses the message and isolates the next parameter.

The value returned is (a reference to) the parameter, unless no further parameters are found (in which case the value "null" is returned).



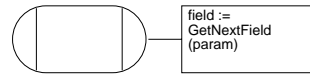
This function searches all records of type "MRec" to find the next one that matches the "ceid" and "sri" parameters.

The value returned can be understood to be a pointer to the record itself, unless no further records are found (in which case the value "null" is returned).



This function searches for all assigned paths of the signalling relationship indicated with the "sri" parameter.

The value returned is the value of an AAL type 2 path identifier, unless no further paths are found (in which case the value "null" is returned).



This function parses the parameter and isolates the next field.

The value returned is (a reference to) the field, unless no further fields exist (in which case the value "null" is returned).



Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 16 de 25)



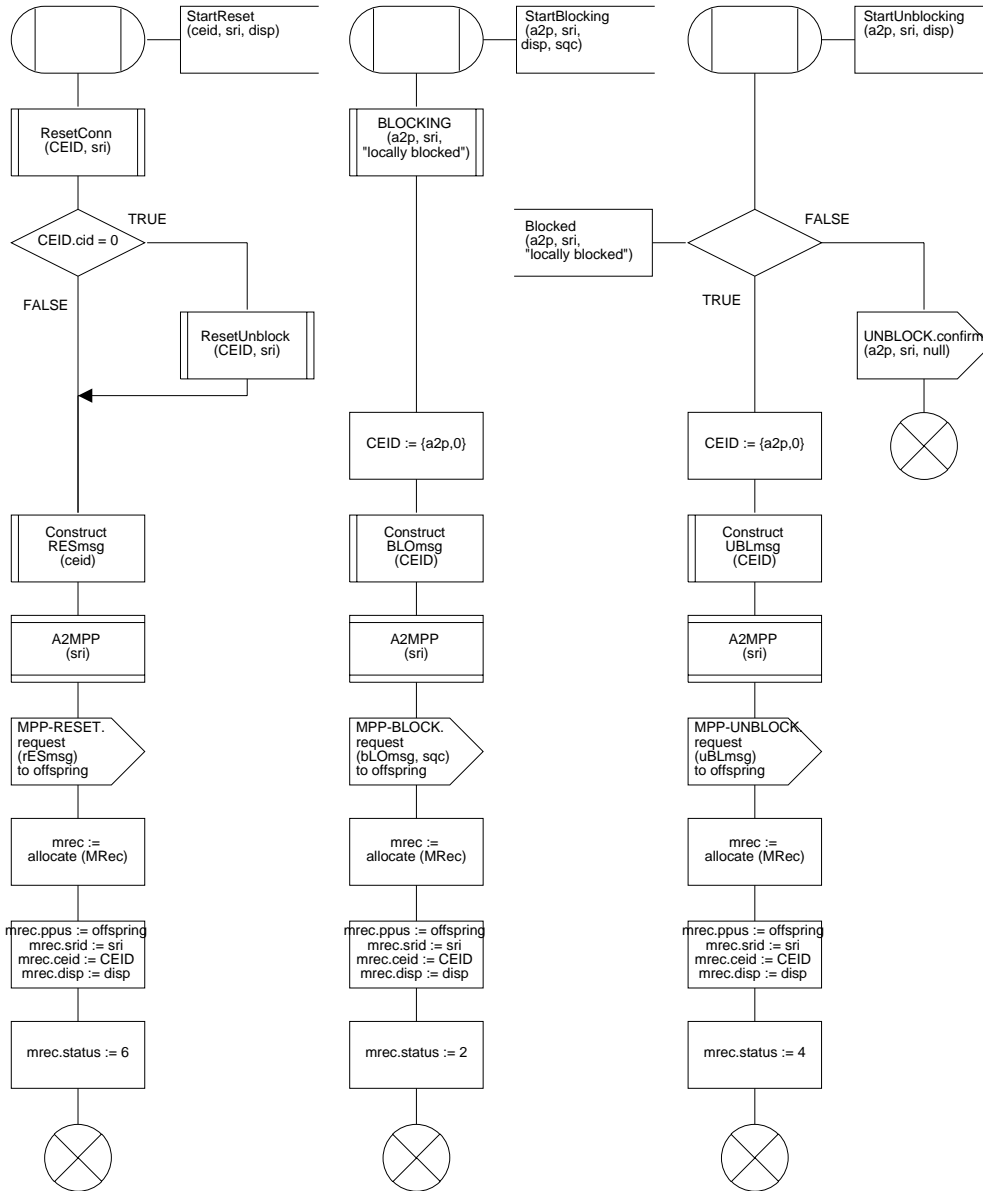


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 17 de 25)

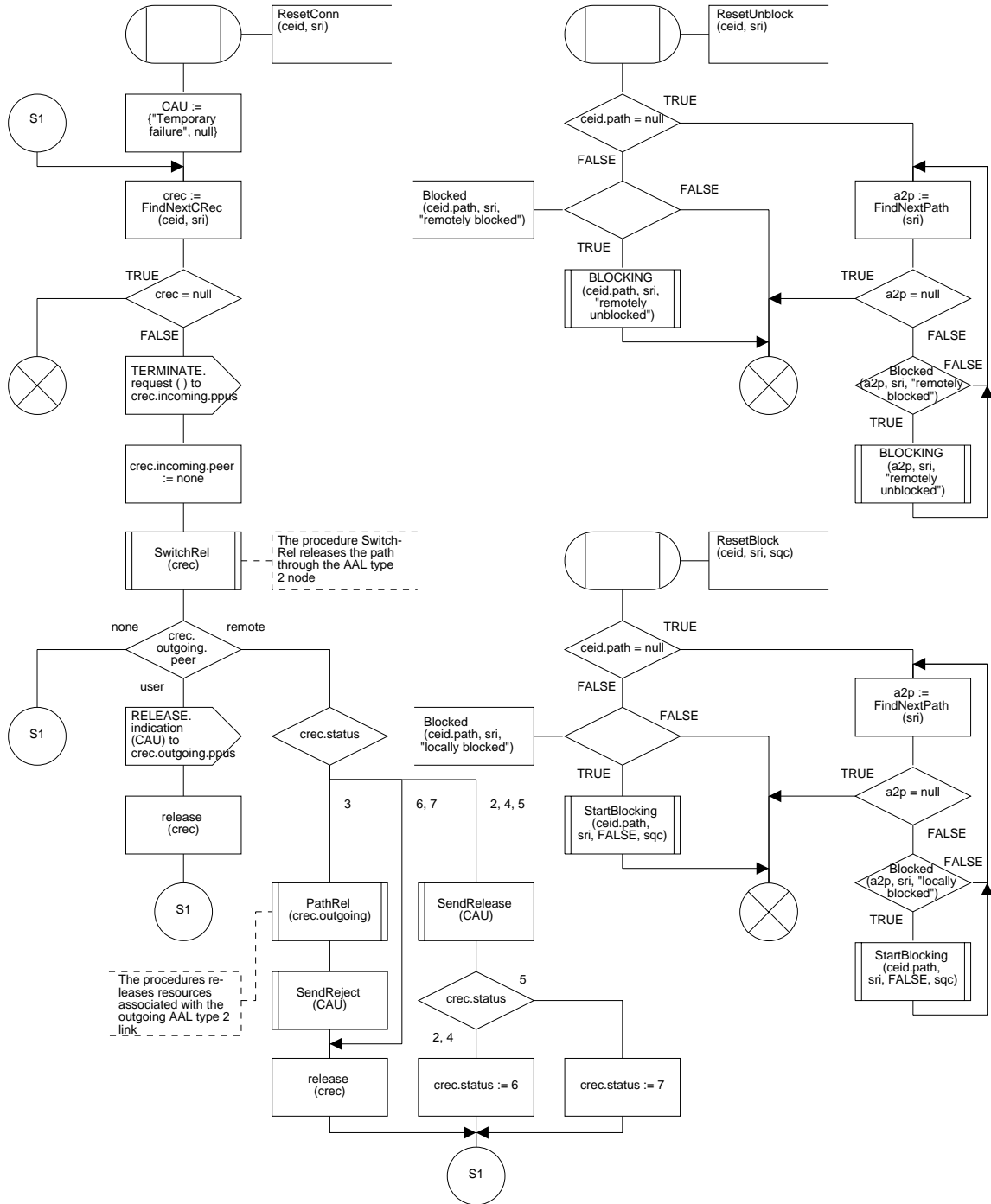


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 18 de 25)

This function checks the compatibility of parameters in recognized messages. It returns a structured value with a course and a cause.

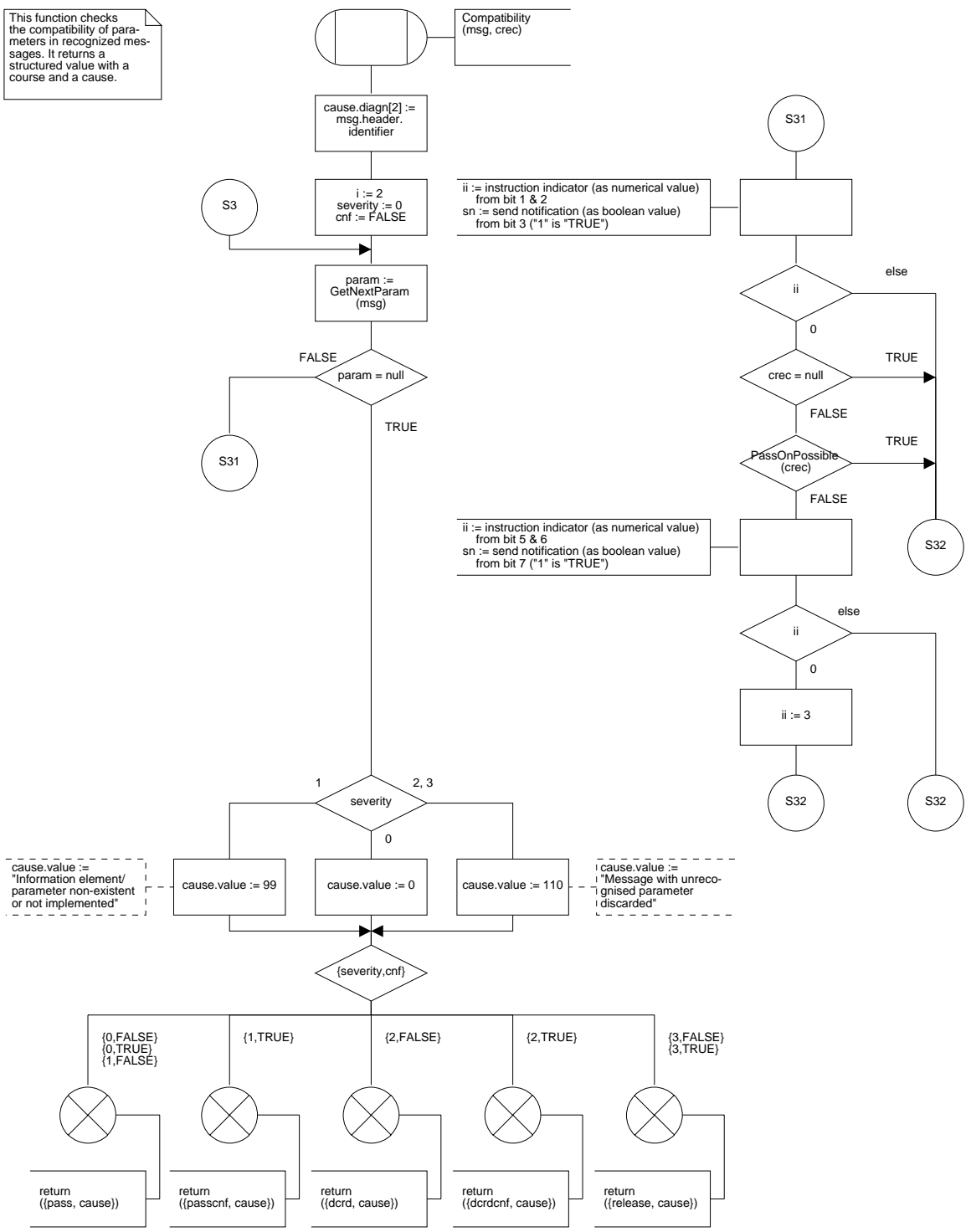


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 19 de 25)

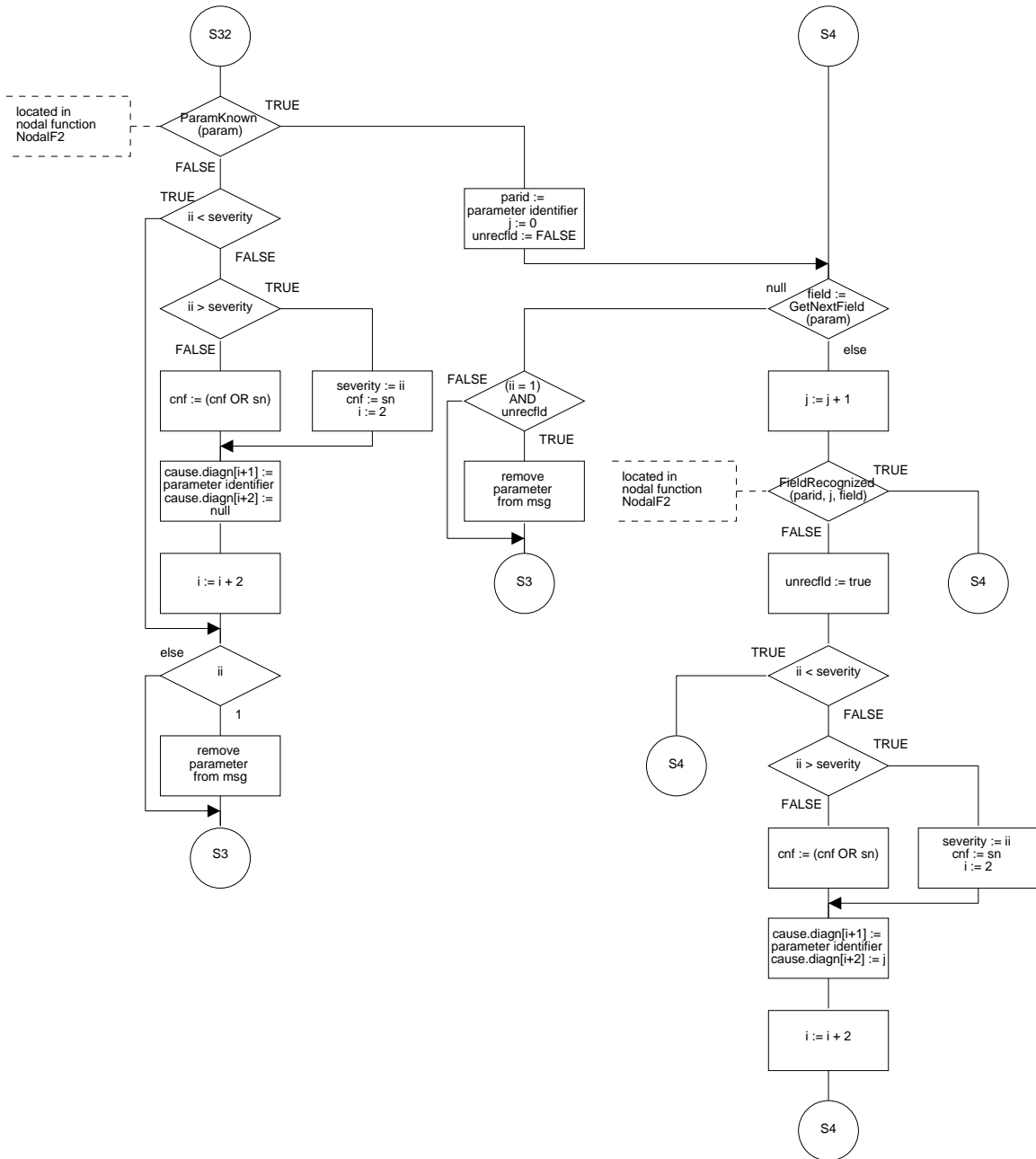


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 20 de 25)

This function checks the compatibility of unrecognized messages. It returns a structured value with a course and a cause.

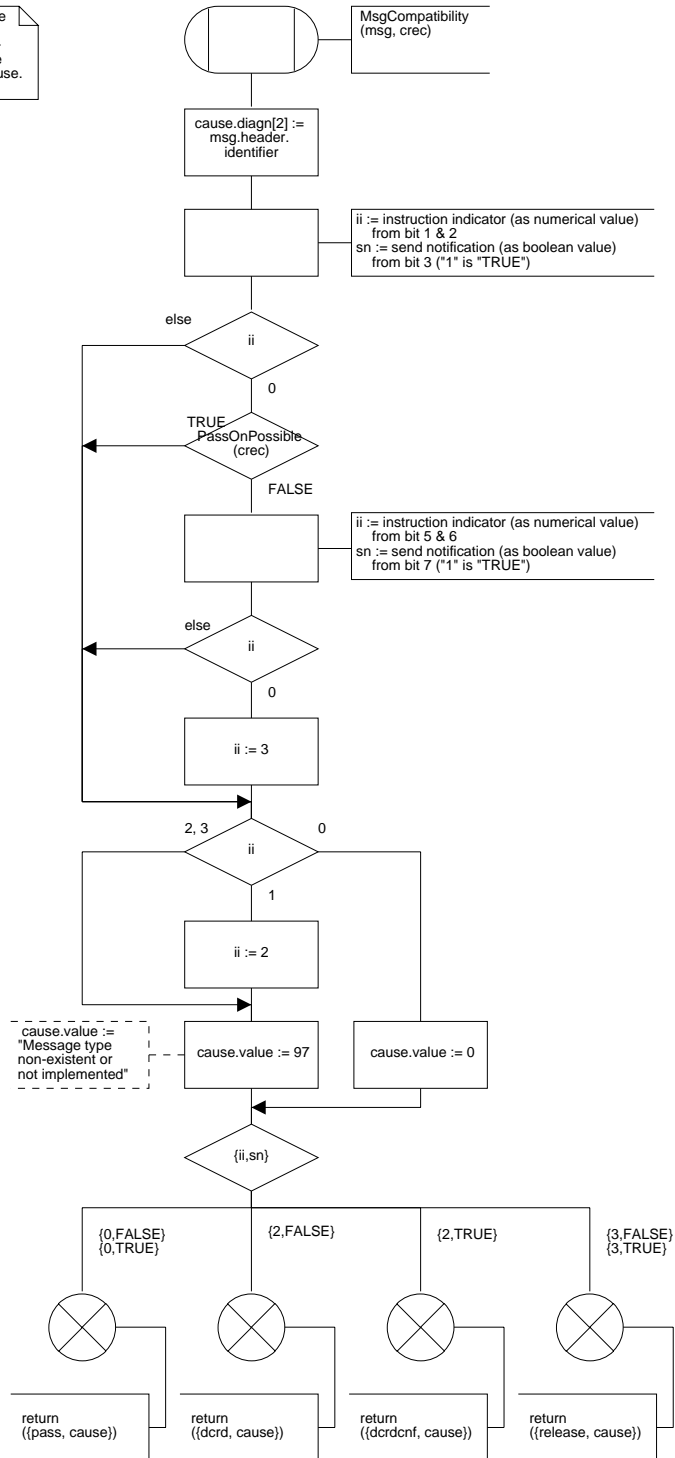


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 21 de 25)

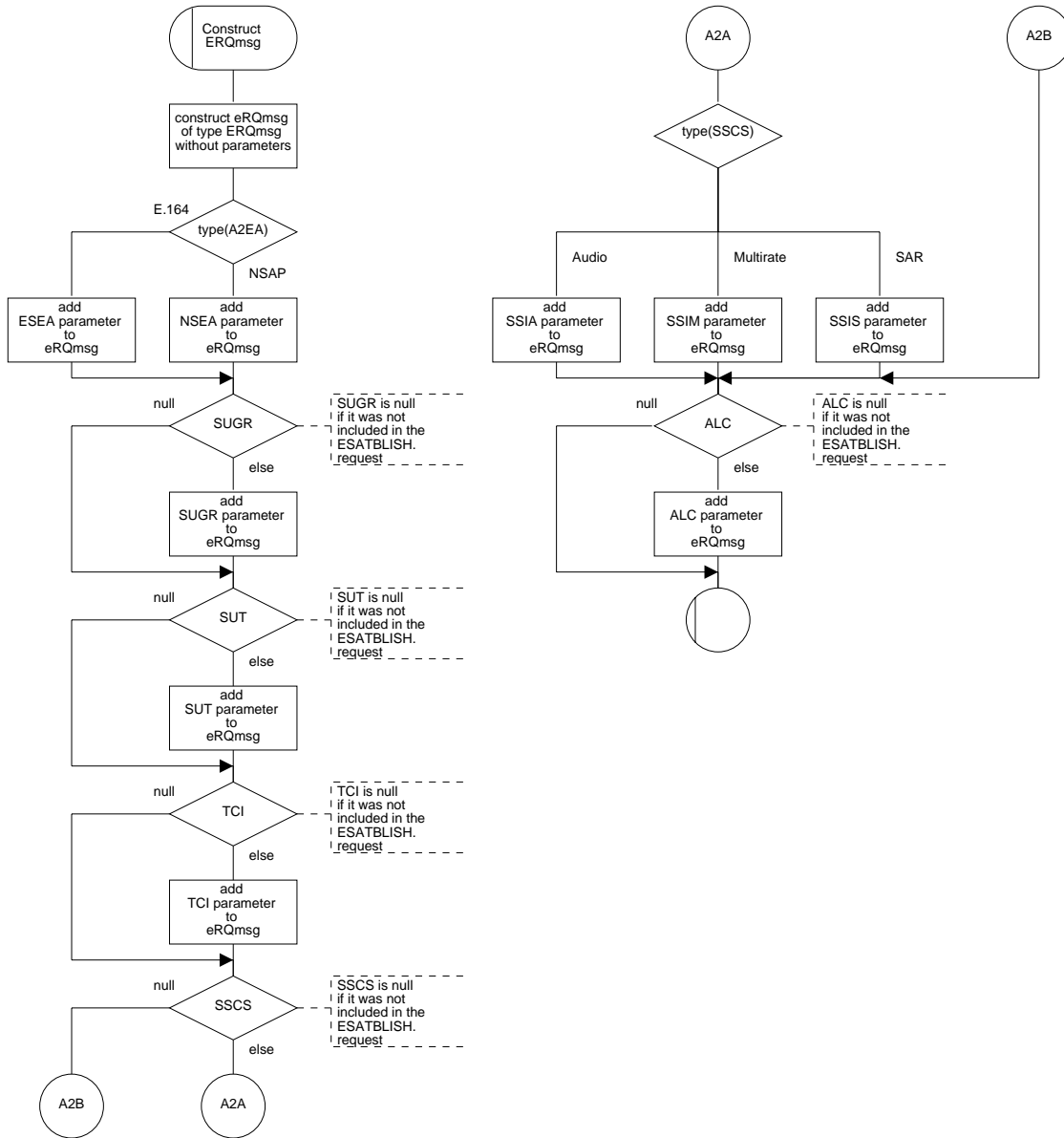


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 22 de 25)

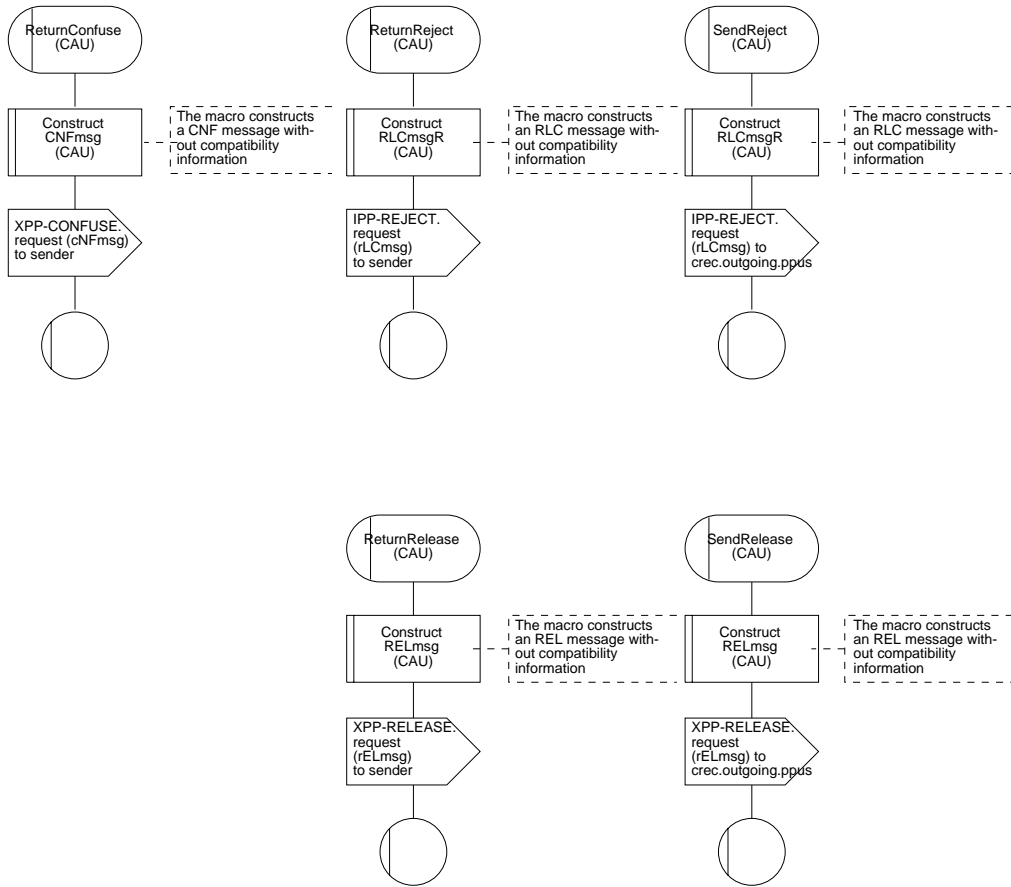


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 23 de 25)

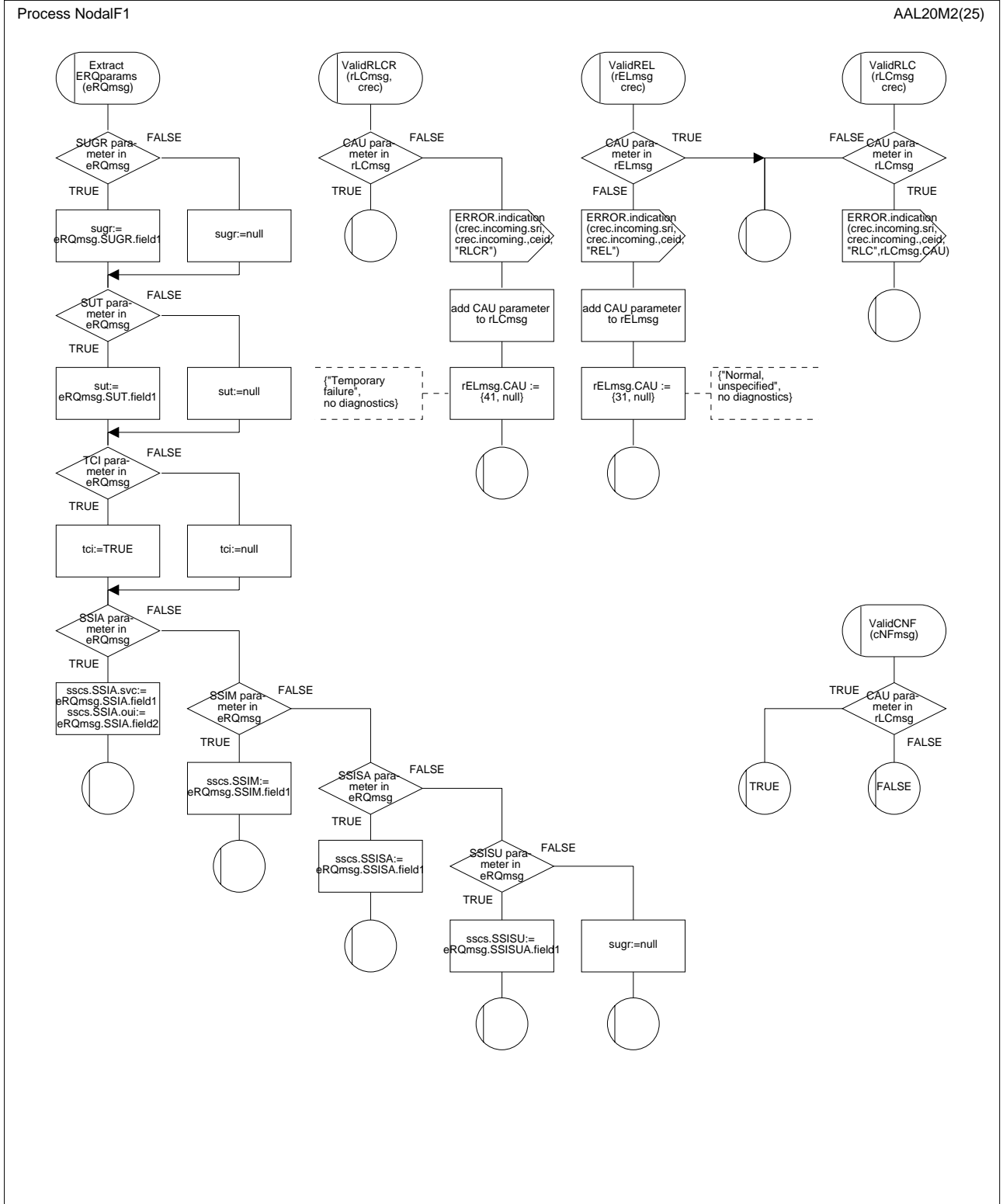


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 24 de 25)



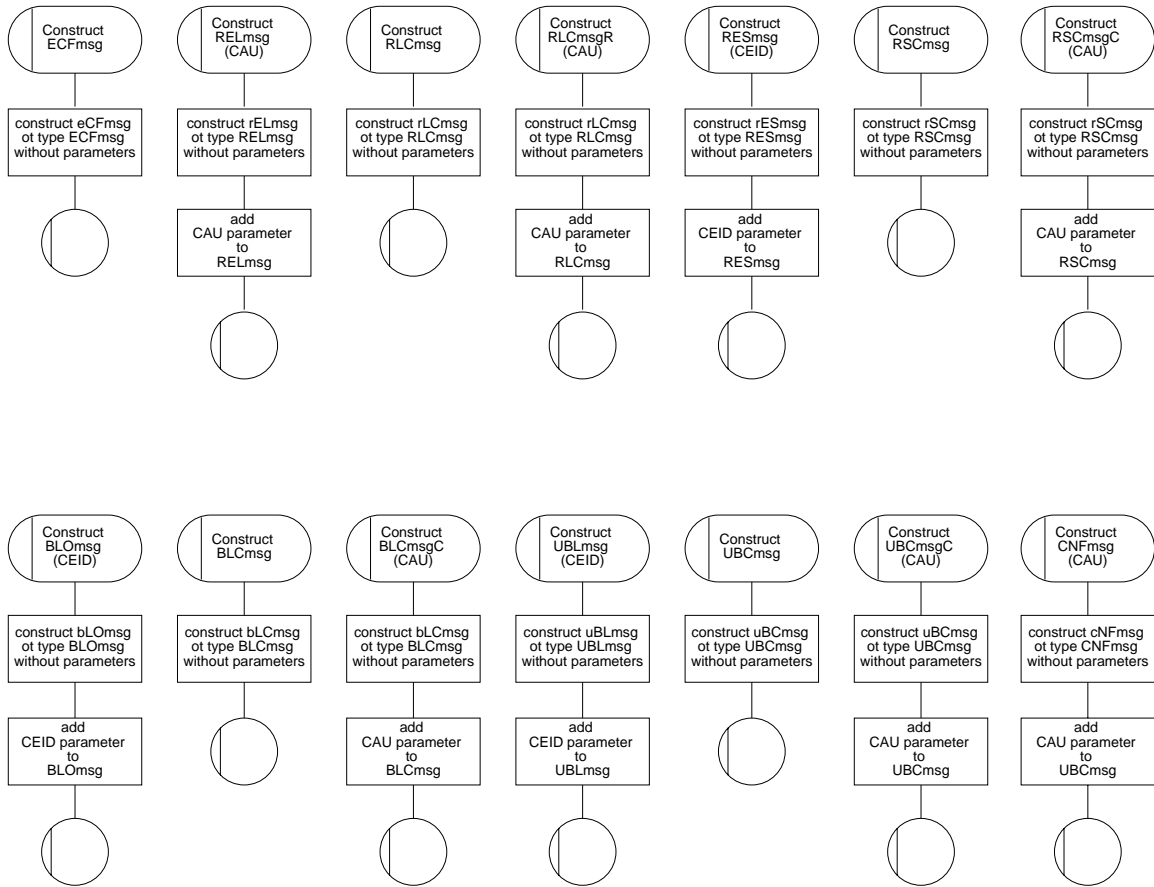


Figura B.3/Q.2630.1 – Diagrama SDL de la función nodal 1 (parte 25 de 25)

## **B.5 Diagramas SDL de las entidades de protocolo**

### **B.5.1 Introducción**

La figura B.2 (parte 3 de 3) muestra que la entidad de protocolo se divide en:

a) *Procedimientos del protocolo saliente*

Esta entidad contiene todas las funcionalidades detalladas en 8.3.2 y, por consiguiente, puede especificarse con precisión mediante diagramas SDL.

b) *Procedimientos del protocolo entrante*

Esta entidad contiene todas las funcionalidades detalladas en 8.3.3 y, por consiguiente, puede especificarse con precisión mediante diagramas SDL.

c) *Procedimientos del protocolo de mantenimiento*

Esta entidad contiene todas las funcionalidades detalladas en 8.3.4 y, por consiguiente, puede especificarse con precisión mediante diagramas SDL.

d) *Interfaz de transporte de señalización*

Esta entidad no se especifica explícitamente en esta Recomendación, sin embargo, es imprescindible para la recepción de primitivas y mensajes procedentes del transporte de señalización y para despacharlos hacia el ejemplar de la entidad de protocolo apropiado y enviar las señales "IN-SERVICE", "OUT-OF-SERVICE", y "CONGESTION" a la función nodal 2. En algunos casos, los ejemplares de entidad de protocolo tienen que ser creados por la interfaz del transporte de señalización (por ejemplo, cuando se reciben los mensajes ERQ, RES, BLO y UBL). Esta entidad se especifica también mediante diagramas SDL.

NOTA – Esta entidad no tiene funcionalidad en el sentido de transmisión.

Así pues, esta cláusula define la entidad de procedimientos del protocolo saliente (véase B.5.2.2), la entidad de procedimientos entrantes (véase B.5.2.3), la entidad de procedimientos de mantenimiento (véase B.5.2.4), y la entidad de interfaz de transporte de señalización (véase B.5.2.5).

### **B.5.2 Diagramas SDL de los procedimientos de protocolo saliente, entrante y de mantenimiento**

#### **B.5.2.1 Estructuras de datos**

Los diagramas SDL de esta cláusula utilizan las definiciones ASN.1 de B.4.2.5.

#### **B.5.2.2 Diagramas SDL de los procedimientos de protocolo salientes**

Los diagramas SDL del procedimiento de protocolo saliente se describen en las partes 1 a 5 de la figura B.4.

#### **B.5.2.3 Diagramas SDL de los procedimientos de protocolo entrantes**

Los diagramas SDL del procedimiento del protocolo entrante se describen en las partes 1 a 5 de la figura B.5.

#### **B.5.2.4 Diagramas SDL de los procedimientos de protocolo de mantenimiento**

Los diagramas SDL del procedimiento de protocolo de mantenimiento se describen en las partes 1 a 5 de la figura B.6.

#### **B.5.2.5 Diagramas SDL de la interfaz de transporte de señalización**

Los diagramas SDL de la interfaz de transporte de señalización se describen en las partes 1 y 2 de la figura B.7.

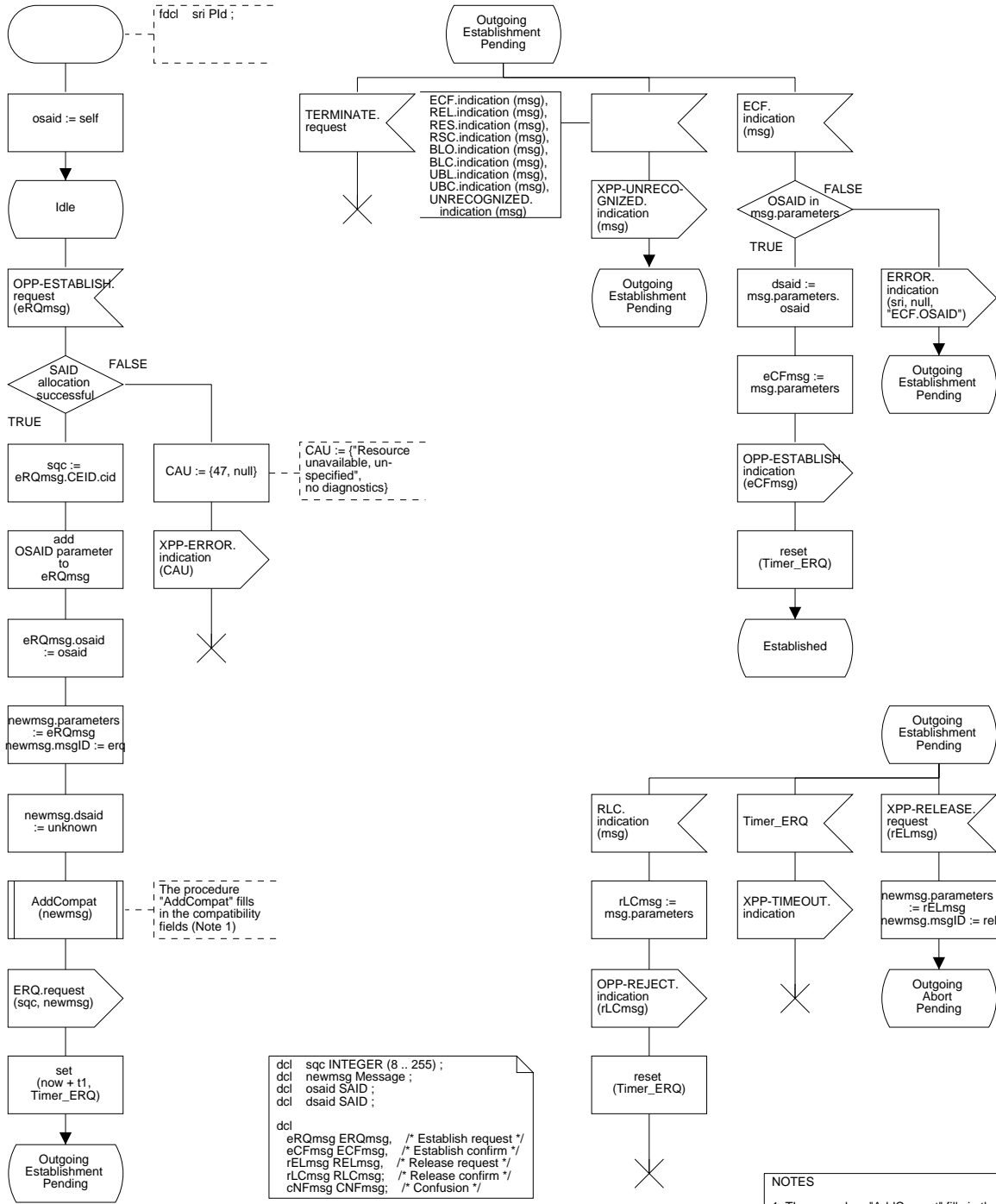
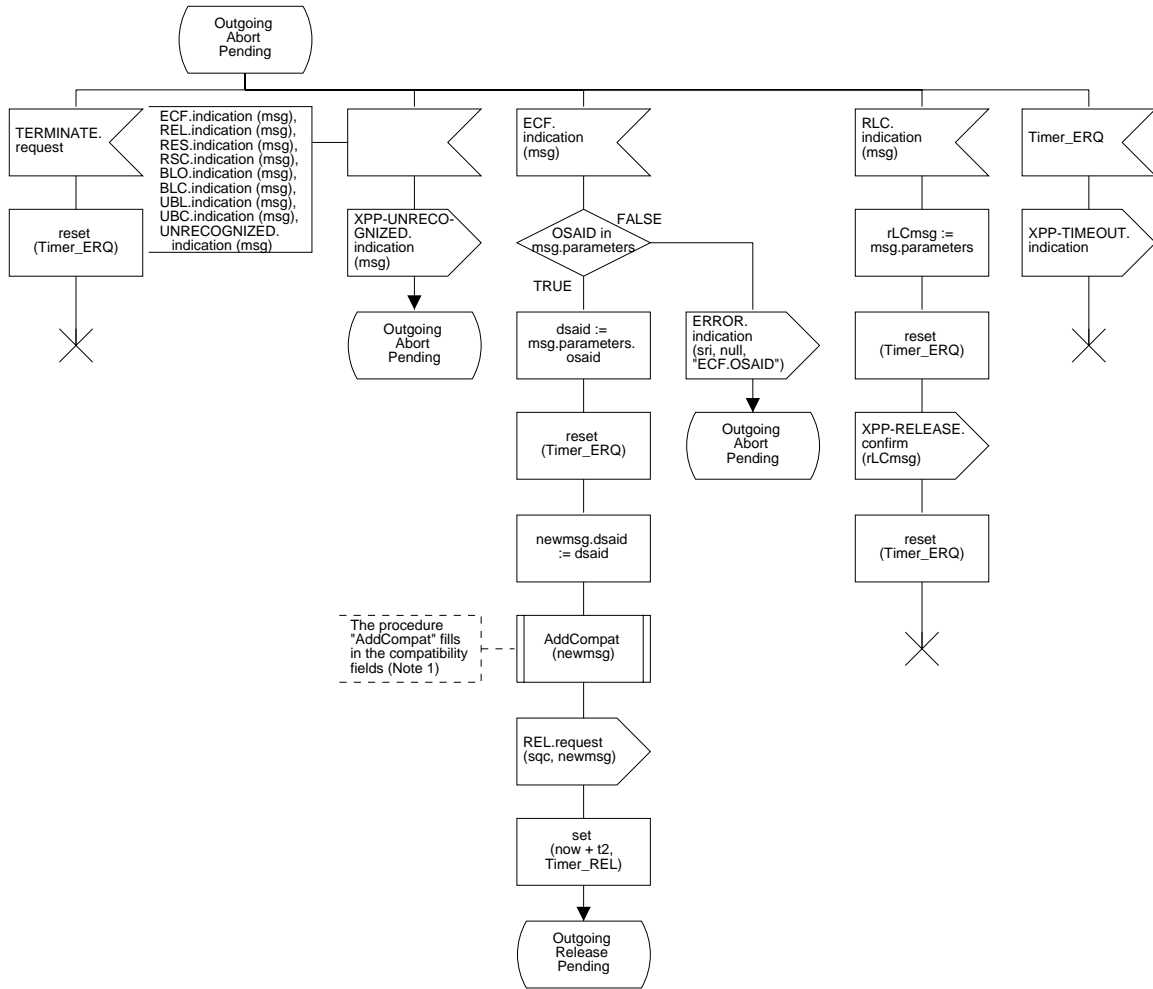


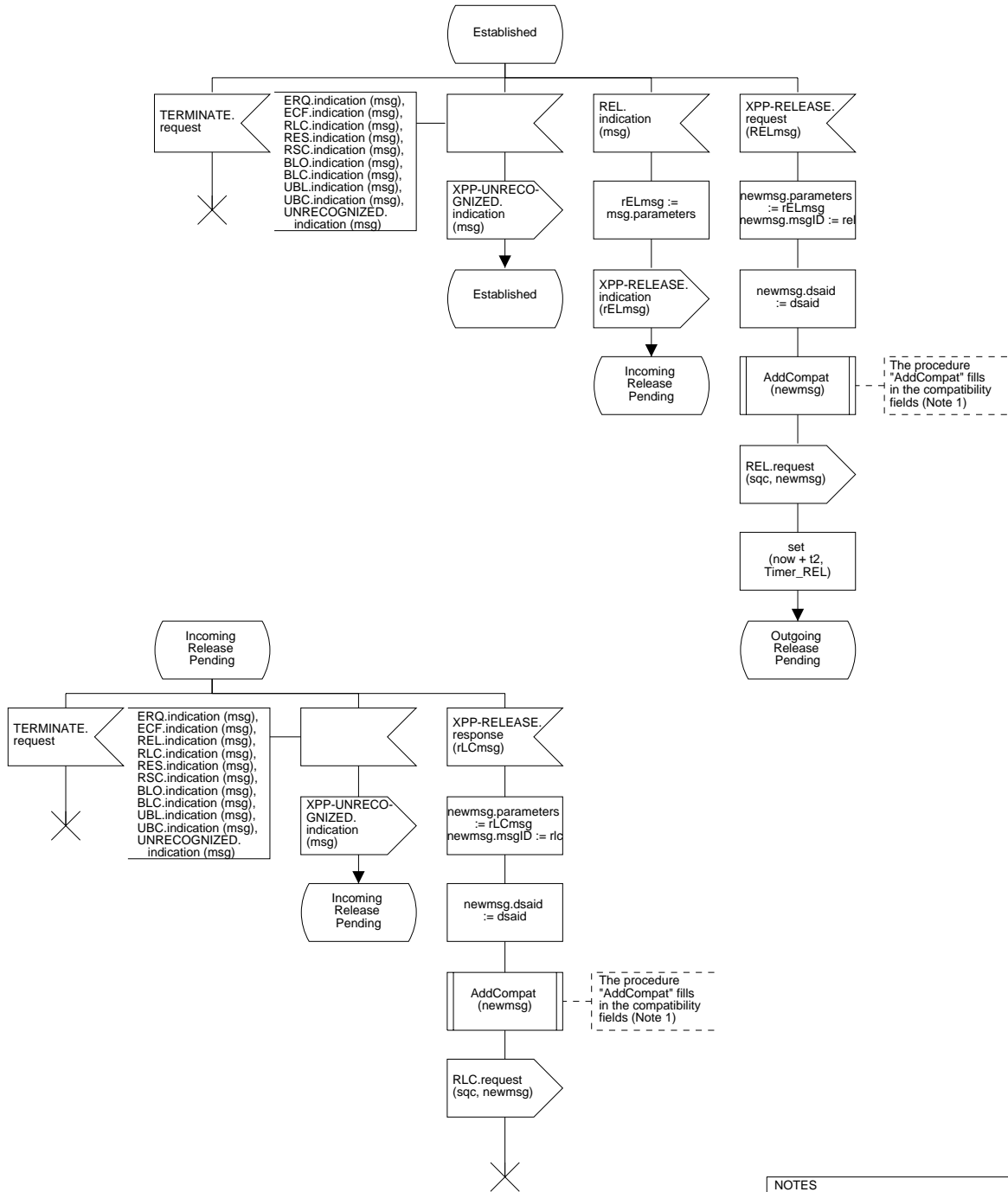
Figura B.4/Q.2630.1 – Diagrama SDL del procedimiento de protocolo saliente (parte 1 de 5)



The procedure "AddCompat" fills in the compatibility fields (Note 1)

NOTES  
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

Figura B.4/Q.2630.1 – Diagrama SDL del procedimiento de protocolo saliente (parte 2 de 5)



**NOTES**  
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

Figura B.4/Q.2630.1 – Diagrama SDL del procedimiento de protocolo saliente (parte 3 de 5)

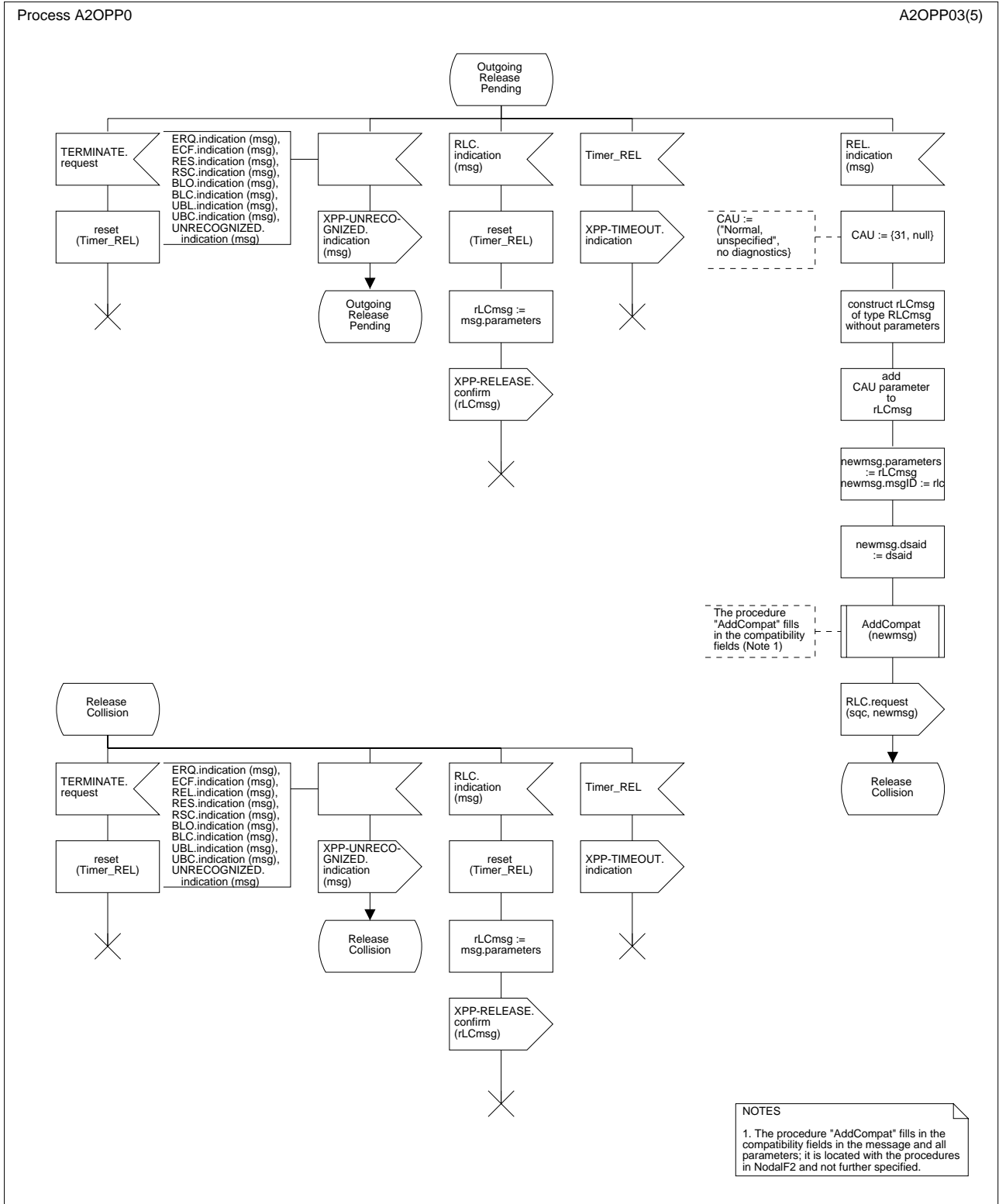
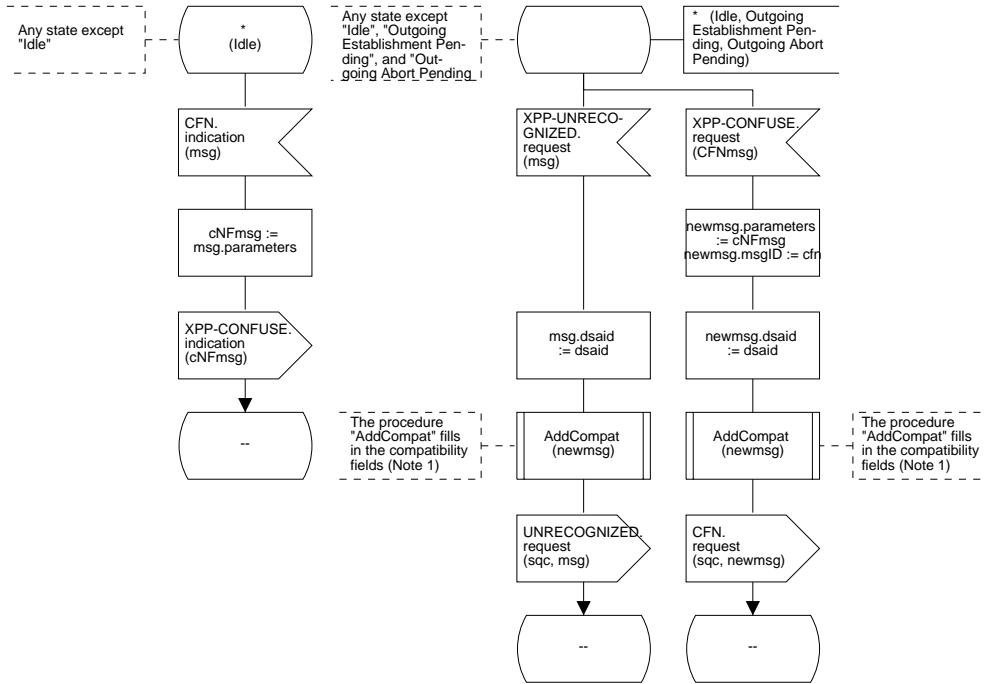
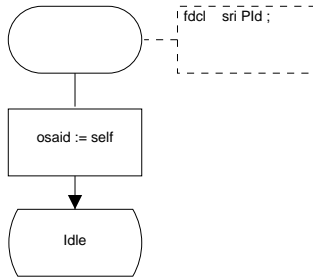


Figura B.4/Q.2630.1 – Diagrama SDL del procedimiento de protocolo saliente (parte 4 de 5)



NOTES  
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

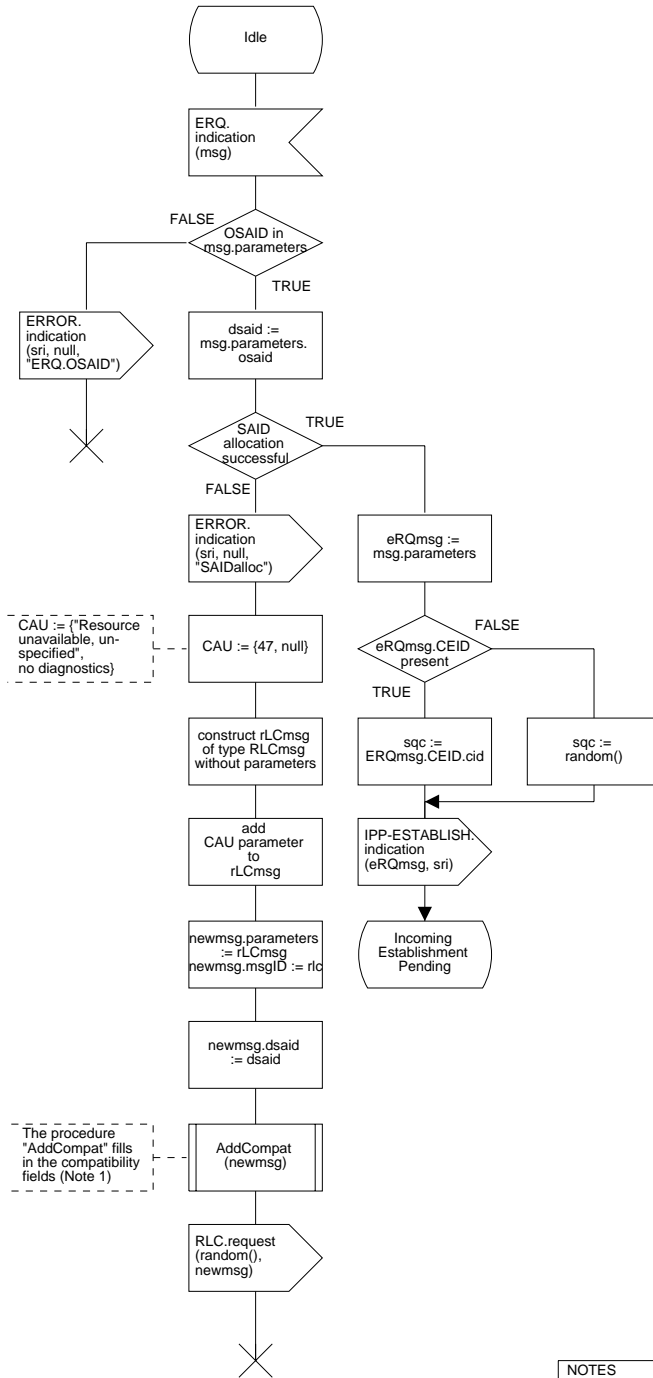
Figura B.4/Q.2630.1 – Diagrama SDL del procedimiento de protocolo saliente (parte 5 de 5)



```

dcl sqc INTEGER (8 .. 255);
dcl newmsg Message;
dcl osaid SAID;
dcl dsaid SAID;

dcl eRQmsg ERQmsg; /* Establish request */
dcl eCFmsg ECFmsg; /* Establish confirm */
dcl rELmsg RELmsg; /* Release request */
dcl rLCmsg RLCmsg; /* Release confirm */
dcl cNFmsg CNFmsg; /* Confusion */
    
```



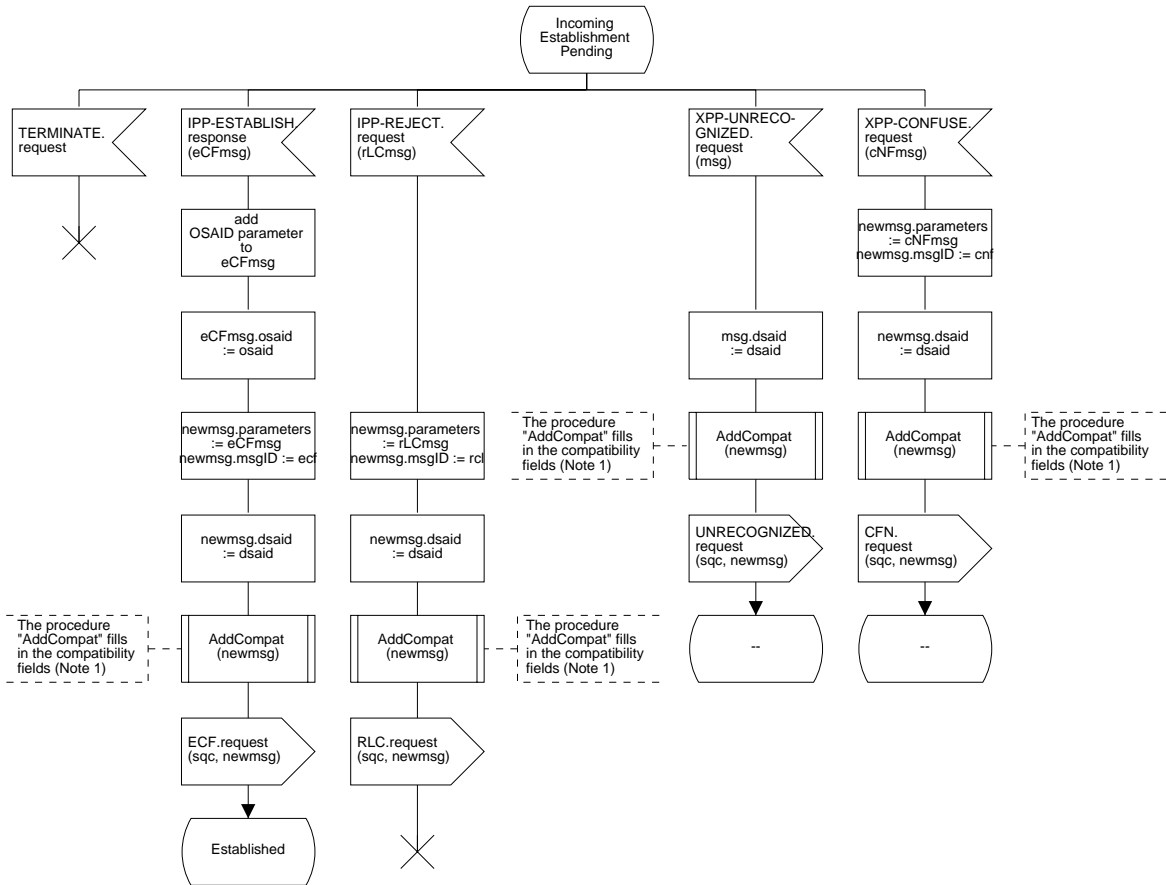
CAU := ("Resource unavailable, unspecified", no diagnostics)

The procedure "AddCompat" fills in the compatibility fields (Note 1)

NOTES  
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

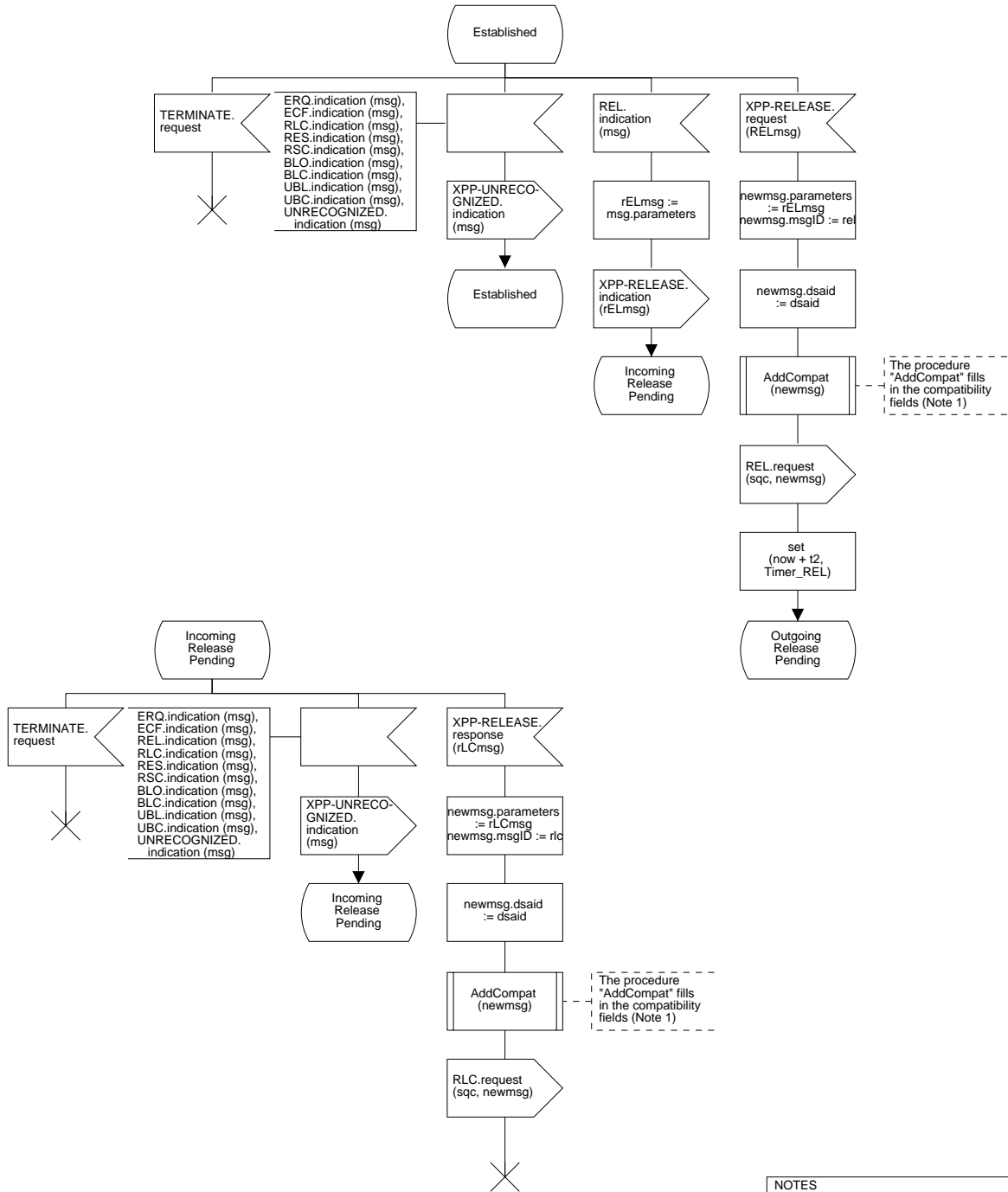
Figura B.5/Q.2630.1 – Diagrama SDL del procedimiento de protocolo entrante (parte 1 de 5)





NOTES  
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

Figura B.5/Q.2630.1 – Diagrama SDL del procedimiento de protocolo entrante (parte 2 de 5)



NOTES  
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

Figura B.5/Q.2630.1 – Diagrama SDL del procedimiento de protocolo entrante (parte 3 de 5)

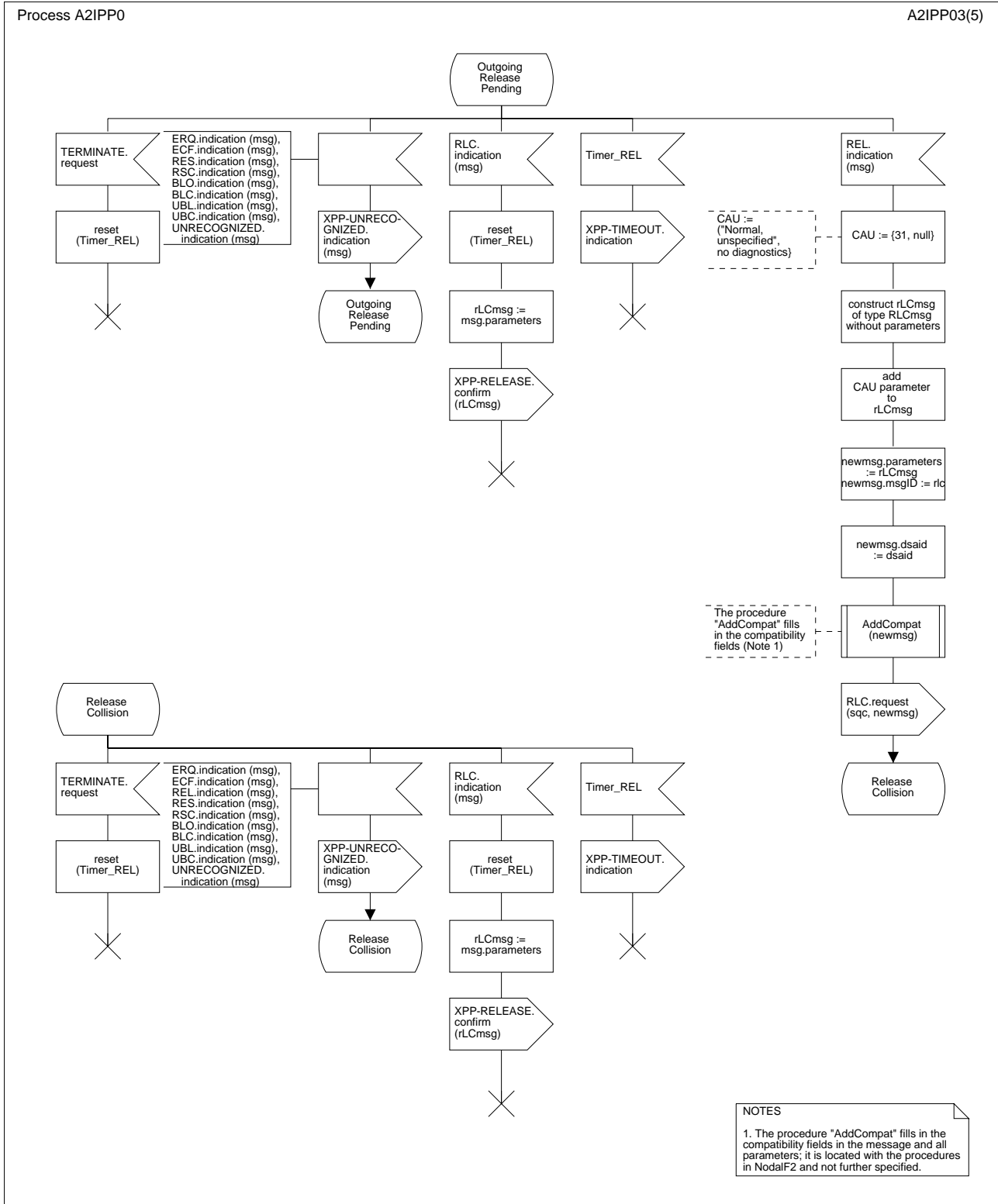
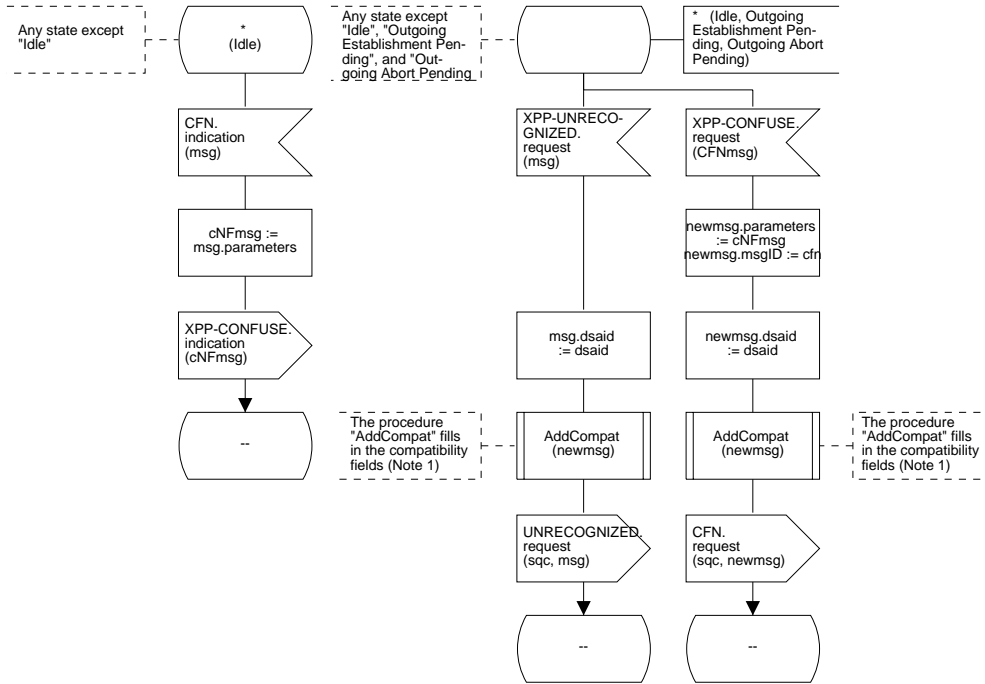
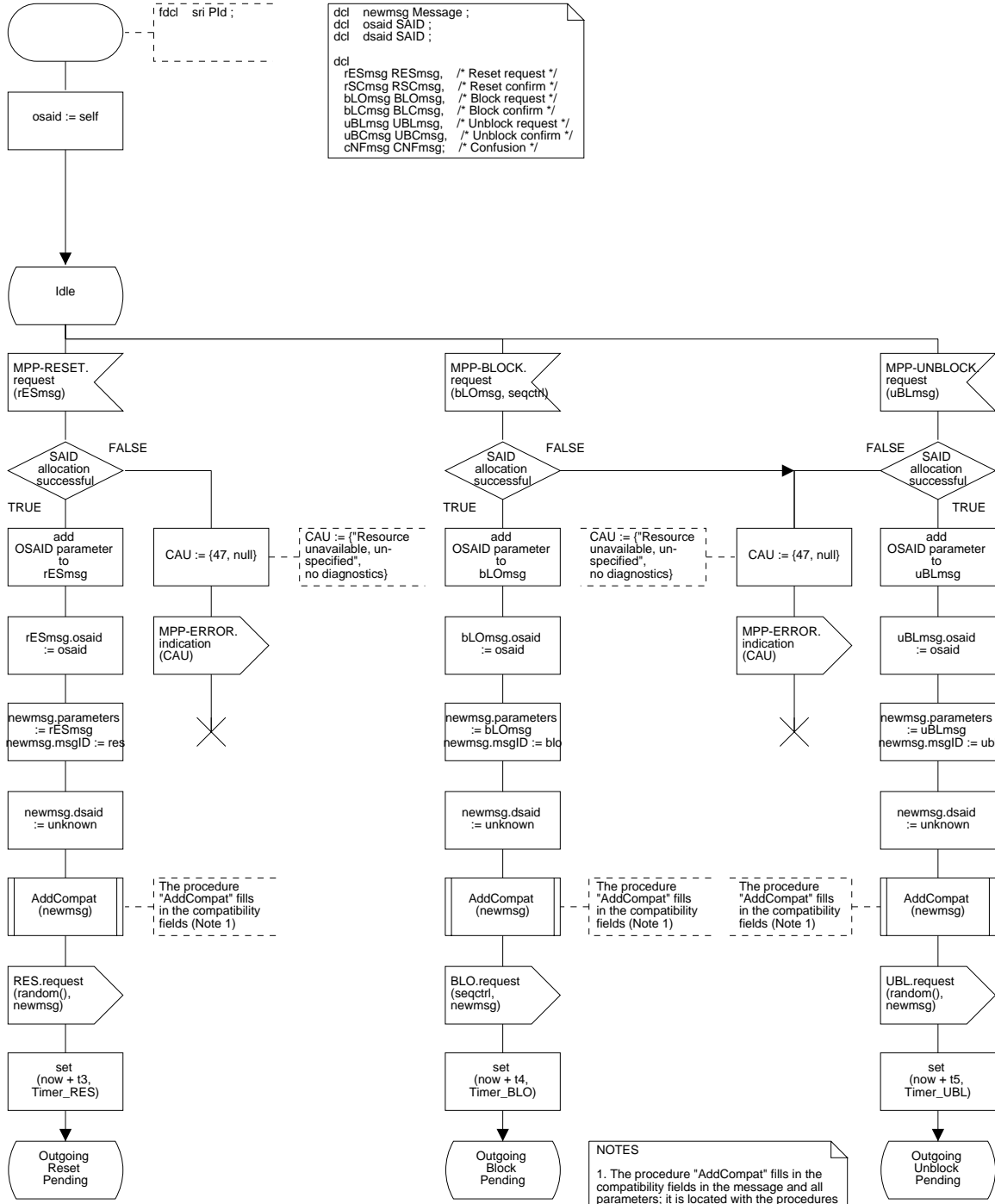


Figura B.5/Q.2630.1 – Diagrama SDL del procedimiento de protocolo entrante (parte 4 de 5)



NOTES  
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

Figura B.5/Q.2630.1 – Diagrama SDL del procedimiento de protocolo entrante (parte 5 de 5)



```

dcl newmsg Message ;
dcl osaid SAID ;
dcl dsaid SAID ;

dcl
rESmsg RESmsg, /* Reset request */
rSCmsg RSCmsg, /* Reset confirm */
bLOmsg bLOmsg, /* Block request */
bLCmsg bLCmsg, /* Block confirm */
uBLmsg uBLmsg, /* Unblock request */
uBCmsg uBCmsg, /* Unblock confirm */
cNFmsg CNFmsg, /* Confusion */
    
```

NOTES  
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

Figura B.6/Q.2630.1 – Diagrama SDL del procedimiento de protocolo de mantenimiento (parte 1 de 5)

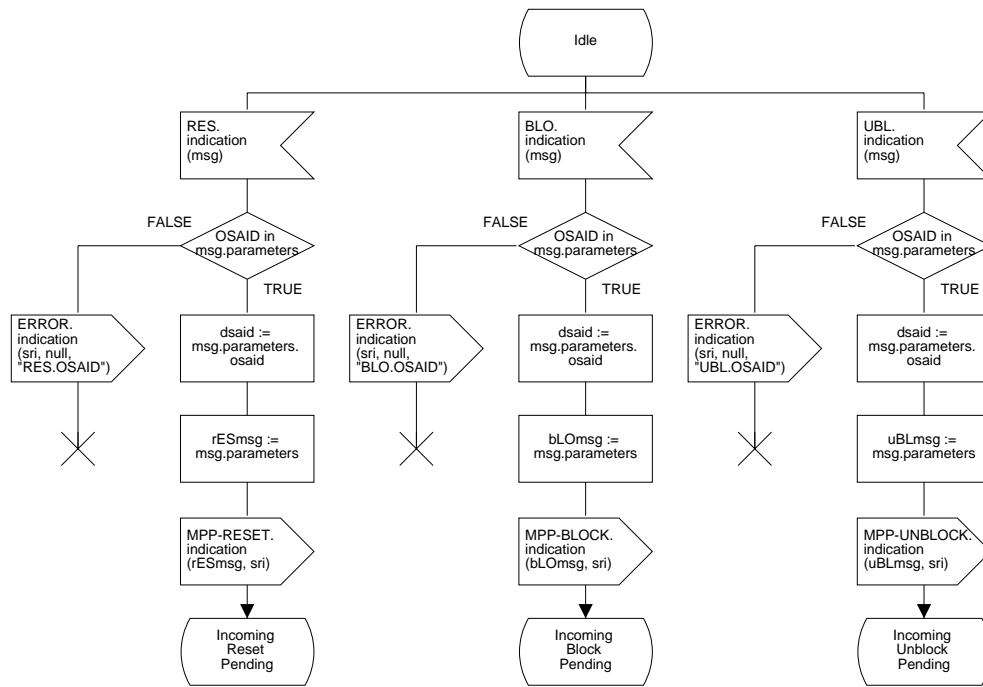


Figura B.6/Q.2630.1 – Diagrama SDL del procedimiento de protocolo de mantenimiento (parte 2 de 5)

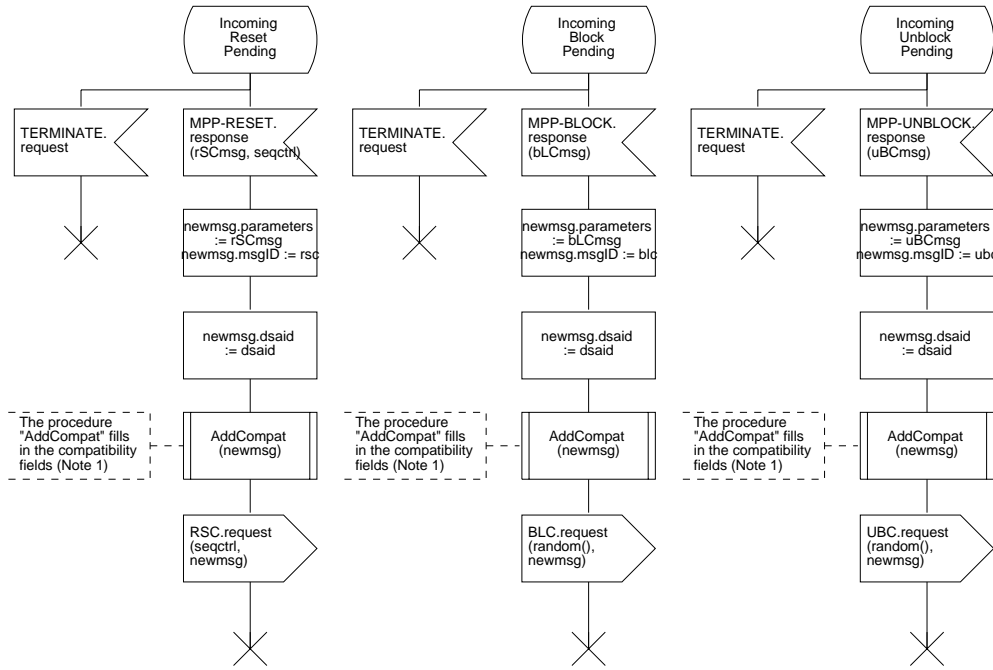
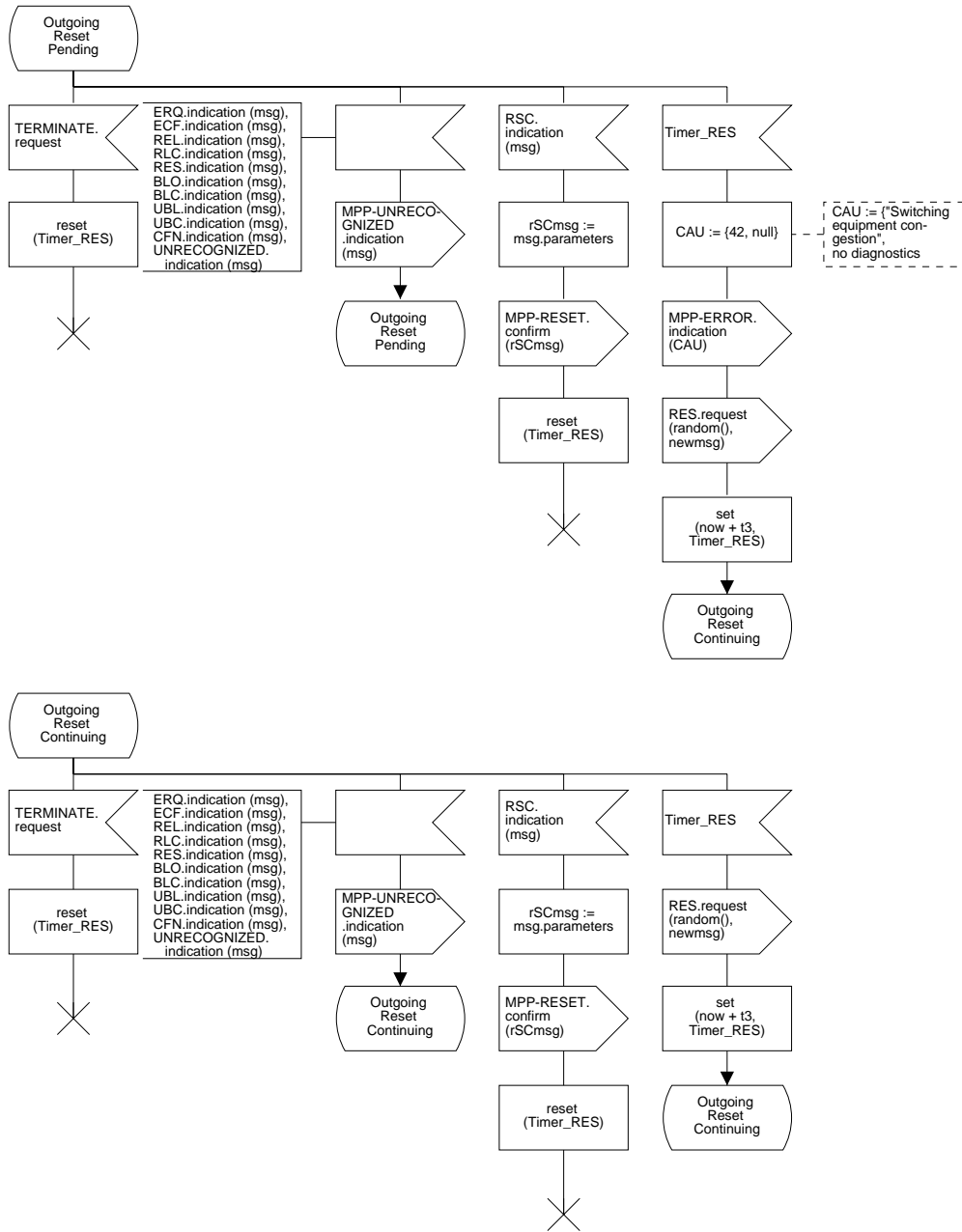


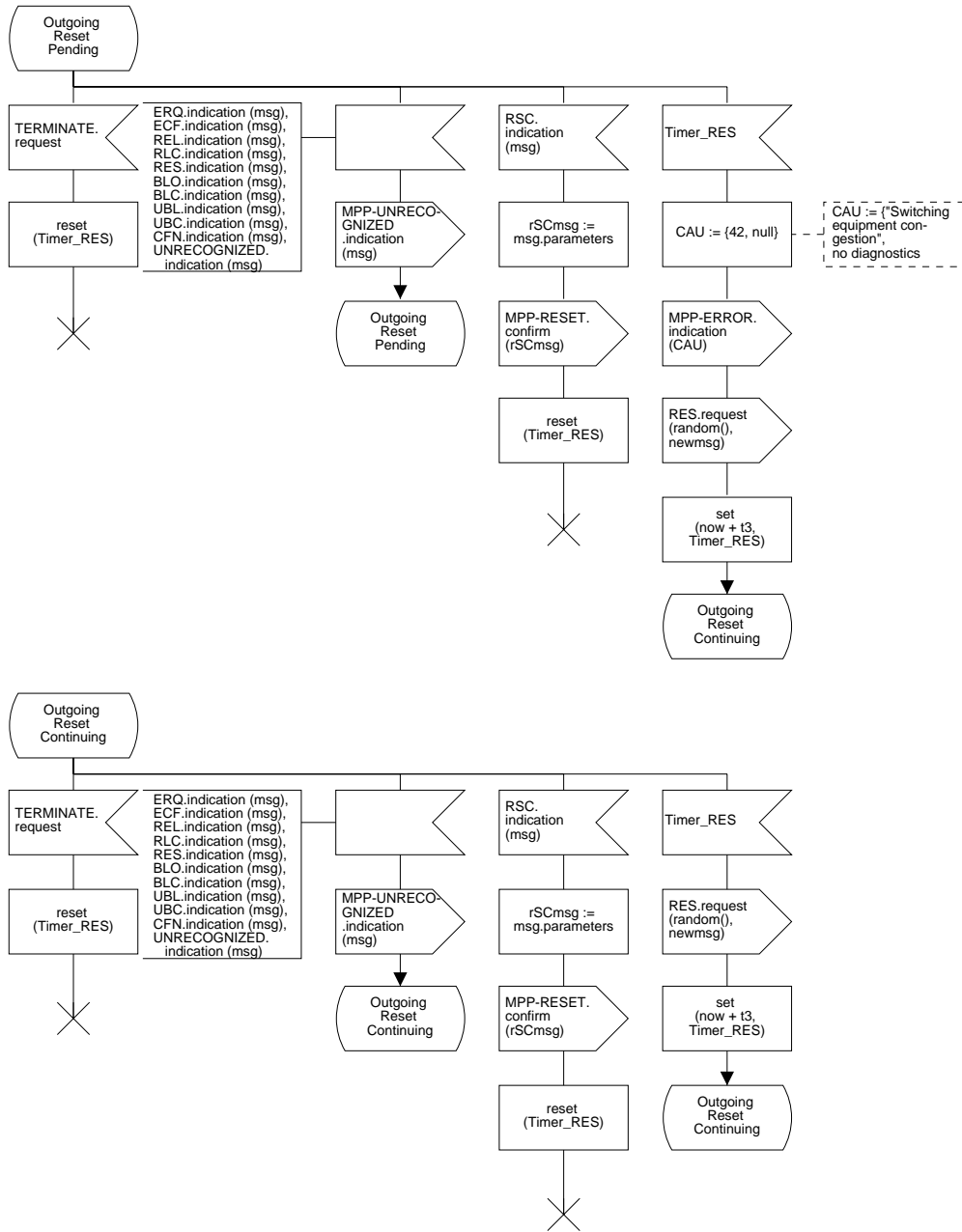
Figura B.6/Q.2630.1 – Diagrama SDL del procedimiento de protocolo de mantenimiento (parte 3 de 5)



**NOTES**  
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

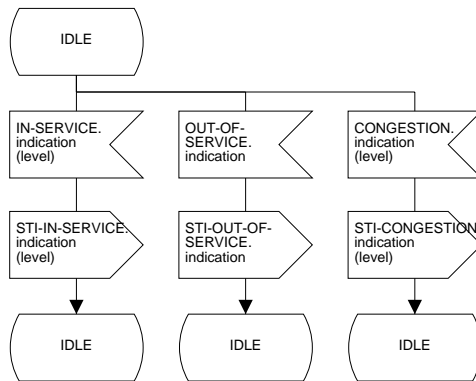
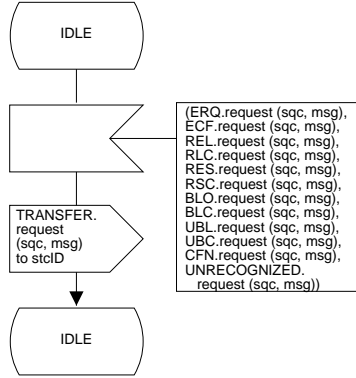
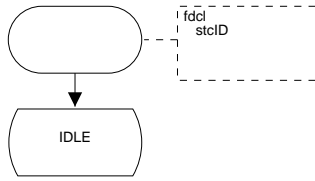
**Figura B.6/Q.2630.1 – Diagrama SDL del procedimiento de protocolo de mantenimiento (parte 4 de 5)**



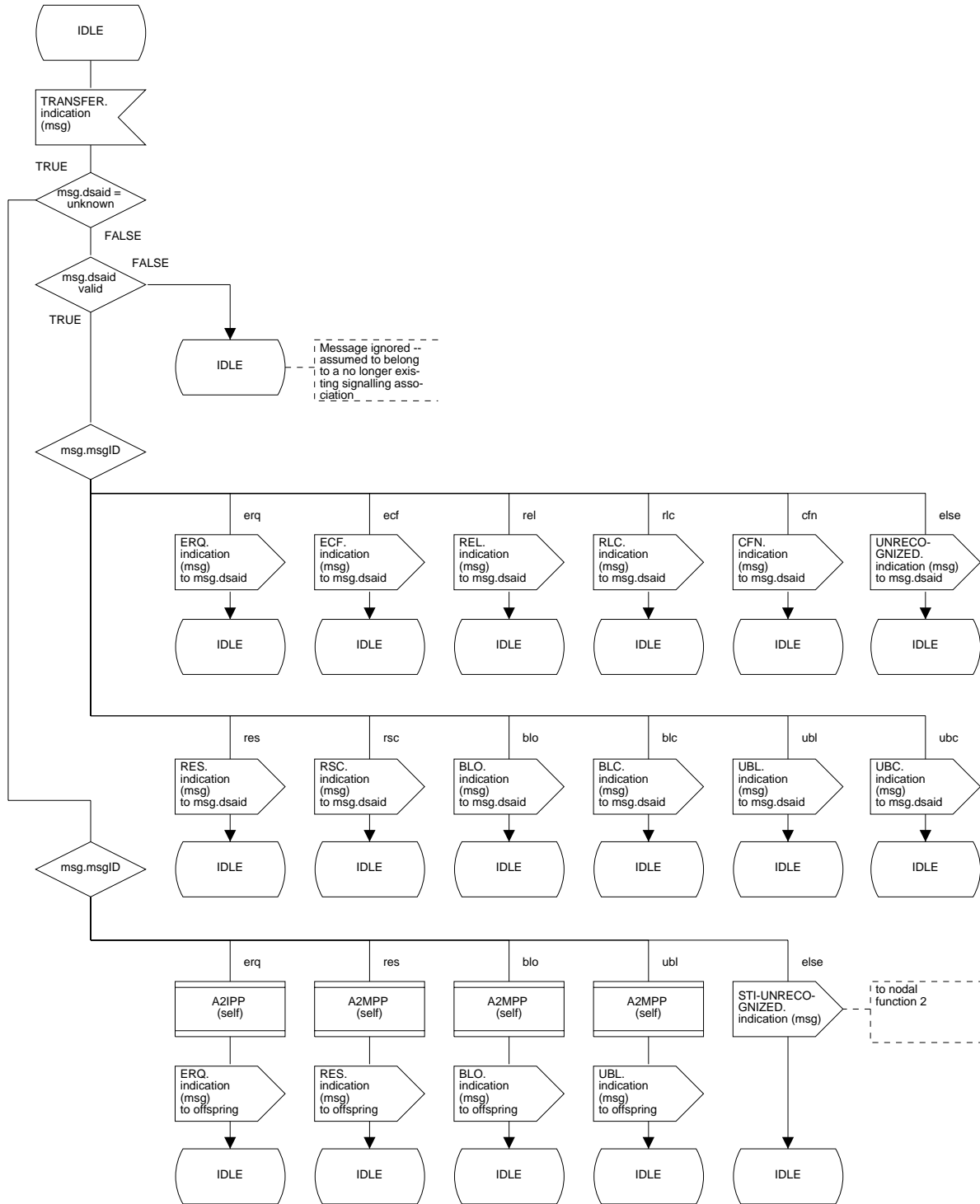


**NOTES**  
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

**Figura B.6/Q.2630.1 – Diagrama SDL del procedimiento de protocolo de mantenimiento (parte 5 de 5)**



**Figura B.7/Q.2630.1 – Diagrama SDL de la interfaz de transporte de señalización (parte 1 de 2)**



**Figura B.7/Q.2630.1 – Diagrama SDL de la interfaz de transporte de señalización (parte 2 de 2)**





## SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedios
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedios
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
<b>Serie Q</b>	<b>Conmutación y señalización</b>
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

**\*20869\***