



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

Q.775

(03/93)

**ESPECIFICACIONES DEL SISTEMA
DE SEÑALIZACIÓN N.º 7**

**SISTEMA DE SEÑALIZACIÓN N.º 7 –
DIRECTRICES PARA LA UTILIZACIÓN
DE CAPACIDADES DE TRANSACCIÓN**

Recomendación UIT-T Q.775

(Anteriormente «Recomendación del CCITT»)

PREFACIO

El Sector de Normalización de las Telecomunicaciones de la UIT (UIT-T) es un órgano permanente de la Unión Internacional de Telecomunicaciones. El UIT-T tiene a su cargo el estudio de las cuestiones técnicas, de explotación y de tarificación y la formulación de Recomendaciones al respecto con objeto de normalizar las telecomunicaciones sobre una base mundial.

La Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se reúne cada cuatro años, establece los temas que habrán de abordar las Comisiones de Estudio del UIT-T, que preparan luego Recomendaciones sobre esos temas.

La Recomendación UIT-T Q.775, revisada por la Comisión de Estudio XI (1988-1993) del UIT-T, fue aprobada por la CMNT (Helsinki, 1-12 de marzo de 1993).

NOTAS

1 Como consecuencia del proceso de reforma de la Unión Internacional de Telecomunicaciones (UIT), el CCITT dejó de existir el 28 de febrero de 1993. En su lugar se creó el 1 de marzo de 1993 el Sector de Normalización de las Telecomunicaciones de la UIT (UIT-T). Igualmente en este proceso de reforma, la IFRB y el CCIR han sido sustituidos por el Sector de Radiocomunicaciones.

Para no retrasar la publicación de la presente Recomendación, no se han modificado en el texto las referencias que contienen los acrónimos «CCITT», «CCIR» o «IFRB» o el nombre de sus órganos correspondientes, como la Asamblea Plenaria, la Secretaría, etc. Las ediciones futuras en la presente Recomendación contendrán la terminología adecuada en relación con la nueva estructura de la UIT.

2 Por razones de concisión, el término «Administración» se utiliza en la presente Recomendación para designar a una administración de telecomunicaciones y a una empresa de explotación reconocida.

© UIT 1994

Reservados todos los derechos. No podrá reproducirse o utilizarse la presente Recomendación ni parte de la misma de cualquier forma ni por cualquier procedimiento, electrónico o mecánico, comprendidas la fotocopia y la grabación en micropelícula, sin autorización escrita de la UIT.

ÍNDICE

Página

1	Introducción	1
1.1	Generalidades	1
1.2	Entorno	1
2	Operaciones.....	2
2.1	Definición	2
2.2	Ejemplos	2
2.3	Facilidades relacionadas con los componentes ofrecidas a los usuarios TC	4
2.4	Situaciones anormales relacionadas con los componentes	8
3	Diálogos	12
3.1	Agrupación de componentes en un mensaje.....	13
3.2	Facilidades de tratamiento de diálogo.....	14
3.3	Facilidades de control de diálogo mejorado	21
4	Directrices para la formulación de especificaciones de protocolos de usuarios TC.....	24
4.1	Introducción.....	24
4.2	Descomposición de la funcionalidad	24
4.3	Especificación de una entidad de aplicación y un contexto de aplicación.....	24
4.4	Especificación de un ASE	25
4.5	Especificación de operaciones y errores.....	25
4.6	Especificaciones de tipos de datos.....	32
4.7	Especificación de sintaxis abstractas	33
4.8	Reglas de codificación.....	34

SISTEMA DE SEÑALIZACIÓN N.º 7 – DIRECTRICES PARA LA UTILIZACIÓN DE CAPACIDADES DE TRANSACCIÓN

(Melbourne 1988, modificada en Helsinki 1993)

1 Introducción

1.1 Generalidades

El objeto de esta Recomendación es proporcionar directrices a los posibles usuarios de capacidades de transacción (usuarios TC). Se facilitan a título ilustrativo solamente, ejemplos que indican la manera en que una aplicación puede utilizar la parte aplicación de capacidades de transacción (TCAP, *transaction capabilities application part*), pero no la manera en que la TCAP debe emplearse en todos los casos. La base técnica de este documento la constituyen las Recomendaciones Q.771 a Q.774, que deben considerarse como referencia primaria en caso de divergencia.

El objeto principal de la TCAP es proporcionar un soporte para aplicaciones interactivas en un entorno distribuido. La TCAP se basa en el concepto de operaciones a distancia definido en las Recomendaciones X.219 y X.229 (ROSE) junto con algunas mejoras y adiciones específicas del entorno del sistema de señalización N.º 7 para proporcionar los servicios que necesitan los usuarios TC. Las interacciones entre las entidades de aplicación distribuida se modelan mediante operaciones. Una operación se invoca mediante una entidad (de origen), en tanto que la otra entidad (de destino) intenta la ejecución de la operación y posiblemente devuelve el resultado de esta tentativa.

La semántica de una operación (representada por su nombre y sus parámetros) no reviste interés para la TCAP, la cual proporciona facilidades independientes de cualquier operación particular. Cuando defina una aplicación, el usuario TC deberá:

- 1) seleccionar operaciones (lo que supone definir la semántica y la sintaxis de los datos intercambiados durante las operaciones de invocación y sus respuestas);
- 2) seleccionar facilidades de la TCAP para soportar esas operaciones. Tales facilidades comprenden el tratamiento de operaciones individuales y la capacidad de disponer de cierto número de operaciones conexas anexadas a una asociación entre usuarios TC, denominada diálogo;
- 3) definir el «guión» (script) de la aplicación (por ejemplo, cuál de los dos pares invoca cada una de las operaciones, el orden de intercambio de mensajes que constituye el diálogo entre los pares de usuarios TC, y sus reacciones ante situaciones anómalas).

En esta Recomendación se describe el proceso de selección para la definición y utilización de las operaciones. Las operaciones que aparecen en lo que sigue son ficticias y se muestran con fines de ilustración solamente. Se describen, asimismo, las facilidades ofrecidas por la TCAP para el tratamiento de una operación o de una secuencia de operaciones en un diálogo. La definición de secuencias específicas de operaciones pertenece a la definición del protocolo de aplicación, que cae fuera del ámbito de esta Recomendación. No obstante, en el Capítulo 4 se proporciona una breve indicación de qué tipo de información debería contener una especificación de aplicación.

Los servicios TCAP son accesibles a los usuarios TC por conducto de primitivas. Estas primitivas modelan la interfaz entre TCAP y sus usuarios, pero no limitan la realización de esta interfaz.

1.2 Entorno

La TCAP define el protocolo de extremo a extremo entre usuarios TC ubicados en una red del sistema de señalización N.º 7. Actualmente no existe ninguna interfaz normalizada para el uso de la TCAP en ninguna otra red o protocolo subyacente (por ejemplo, X.25) distinto del SCCP del sistema de señalización N.º 7.

TC considera los usuarios que son sensibles al tiempo real y que no necesitan intercambiar grandes cantidades de datos. Se considera que para este tipo de usuarios, los protocolos normalizados definidos por las capas 4 a 6 de la OSI en las Recomendaciones de la serie Rec. X producirán taras (overheads) excesivas, por lo que no se utilizan.

En consecuencia, la TCAP no puede soportar todas las clases de aplicaciones y algunas de éstas necesitarán apoyarse en servicios más elaborados, tales como los especificados en las Recomendaciones de la serie X. Además de expresar lo que puede efectuar la TCAP, esta Recomendación indica lo que es incapaz de realizar, a fin de ayudar al diseñador de aplicaciones a elegir la manera de soportar una aplicación.

2 Operaciones

2.1 Definición

Un usuario TC de origen invoca una operación para pedir a un usuario TC de destino que realice una acción determinada.

Cada operación pertenece a una clase particular, que indica si el destino debe o no comunicar resultado con éxito (outcome), sin éxito (error), o ambos, o ninguno.

La clase de una operación no se señala al usuario TC distante cuando se invoca la operación; se supone que en ambos extremos la aplicación posee una comprensión común de la clase de cada operación en curso.

Además de la clase, la definición de una operación incluye un valor de temporización que indica el tiempo máximo en el que debe completarse la operación y el resultado comunicado.

Este valor de temporizador es un asunto local. No se comunica al extremo distante a través de ningún protocolo. Lo elige el usuario TC cuando define la operación en base a las expectativas del tiempo de propagación de ida y retorno de un usuario TC a otro y de las demoras del procesamiento.

Una operación se define por:

- su código de operación y el tipo de los parámetros asociados con la petición de operación;
- su clase;
- si la clase requiere un informe de éxito, los posibles resultados correspondientes a las ejecuciones exitosas se definen mediante una lista de parámetros;
- si la clase requiere un informe de fallo, los posibles resultados correspondientes a situaciones en que el usuario TC distante no pudo ejecutar completamente la operación. Cada una de estas situaciones se identifica mediante una causa de error específica; la lista de estas causas de error forma parte de la definición de la operación. Puede añadirse información de diagnóstico a la causa de error: si está presente, forma parte de la definición;
- la lista de las posibles operaciones enlazadas (linked operations), si se admiten respuestas consistentes en operaciones enlazadas para esa operación. Las operaciones enlazadas deberán describirse separadamente;
- un valor de temporizador que indique el intervalo en el cual debe completarse la operación y devolverse, caso de existir. Este valor de temporización puede ser uno de los factores utilizados por una realización para gestionar el ID de invocación asociado a la invocación de la operación. (Cuando expira el temporizador asociado a una invocación de operación, se devuelve el ID de invocación al conjunto de ID de invocación tras un periodo de «congelación» adecuado que depende de la realización.)

Por regla general, la elección de la clase de operación debe basarse en la semántica asociada a la operación, y las operaciones no deben tener que cursarse en un tipo de mensaje en particular. Por ejemplo, si el invocador de la operación no requiere un acuse de que ésta ha podido llevarse a cabo o no, una operación de clase 4 puede ser la más apropiada. Si el invocador de una operación no necesita un conocimiento explícito de que la misma se ha realizado con éxito pero desea saber si no se la ha podido realizar en absoluto, es conveniente utilizar una operación de clase 2. Por ejemplo, la definición de la operación «Anuncio» como de clase 2 ó 4 sugiere distintas intenciones del invocador de la operación, aunque las acciones del realizador de la operación sean idénticas.

2.2 Ejemplos

2.2.1 Tratamiento de operaciones simples

Nota – La invocación de la operación, así como el Informe de resultado con éxito debe caber en un mensaje. Los Informes de éxito pueden segmentarse utilizando (retorno) de resultado-no último y (retorno) de resultado-último.

Clase 1 (se informan tanto el éxito como el fallo)

Traducir un número con franquicia a un número de abonado llamado; devolver el número llamado si puede efectuarse la traducción. En otro caso indicar por qué no es así; tiempo atribuido: 2 segundos.

Si no se devuelve una respuesta a una invocación de operación una vez expirado el temporizador, se informa al usuario TC (cancelación de la operación por la TCAP); éste puede suponer que la invocación o la respuesta se ha perdido y, según los requisitos de la operación, tomar las medidas correctivas adecuadas (por ejemplo, invocar nuevamente la operación, informar a la gestión local, etc.).

Clase 2 (sólo se informa el fallo)

Realizar una prueba de rutina y enviar una respuesta solamente en el caso en que exista algún fallo; tiempo atribuido: 1 minuto.

En el caso de una operación de clase 2, si no se recibe ningún resultado cuando expira el temporizador, se informa al usuario TC. Esto se interpreta como un resultado con éxito, aun cuando se pierda la invocación.

Cuando se seleccione la clase 2 deberá contemplarse este aspecto.

Clase 3 (sólo se informa el éxito)

Realizar una prueba: esto corresponde a un punto de vista pesimista en que se considera el fallo como opción por defecto que no requiere ninguna respuesta.

Se informa al usuario TC sobre la expiración del temporizador: el usuario TC deberá interpretar esta información como un fallo de la operación (aunque esto es considerado normal por el TC, que estima que la operación ha terminado). Al seleccionar la clase 3 deberá considerarse este aspecto.

Clase 4 (no se informa ni el éxito ni el fallo)

Enviar un aviso, sin esperar una respuesta ni acuse de recibo de ningún tipo.

En este caso, no se desprende ningún resultado de la invocación de la operación. El usuario TC confía en la TCAP y en la red para la entrega de la invocación. La notificación de la expiración del temporizador es un asunto local.

Comparación con las clases de operación ROSE (Recomendación X. 219)

El ROSE establece cinco clases de operaciones: las clases 2 a 5, denominadas clases asíncronas, son idénticas a las clases 1 a 4 de la TCAP. La clase 1 del ROSE es una clase síncrona que no tiene correspondencia en la TCAP, donde se consideran intercambios dúplex de componentes.

Sin embargo, un usuario TC puede decidir operar en un modo síncrono (véase 3.2.1).

2.2.2 Tratamiento de operaciones más complejas

Operaciones con resultados segmentados

Un resultado exitoso puede dividirse en varios segmentos, cada uno de los cuales se indica al originador de la aplicación mediante una primitiva. Los usuarios TC pueden emplear esta facilidad, que utiliza la primitiva TC-RESULTADO-NL, para remediar la ausencia de segmentación en las capas subyacentes. El último segmento se indica mediante la primitiva TC-RESULTADO-L.

El informe de un error no puede ser segmentado.

Cuando el diseñador del protocolo puede asegurar que la segmentación la proporcionan las capas subyacentes a través de las rutas de señalización por las que se transfieren los mensajes TC, se desaconseja el empleo de esta facilidad de segmentación.

En el caso de un resultado segmentado, la TC no puede identificar un segmento específico.

El usuario TC debe asegurar que se pueda pasar cada segmento (es decir, el parámetro Parámetro de cada primitiva petición TC-RESULTADO-NL/L debe contener información suficiente como para permitir la construcción de un valor válido del tipo (o subtipo compatible) asociado con el resultado de la operación.

Ejemplo E1: Una operación solicita la ejecución de una prueba. El resultado de una ejecución correcta se segmenta en tres partes P1, P2 y P3, que se devuelven al originador.

Una posible secuencia de primitivas para el ejemplo E1 se indica en el Cuadro 1.

Operaciones enlazadas

Otro elemento del método de operación básico es la capacidad de enlazar una invocación de operación con otra invocación de operación.

Normalmente esta facilidad abarca situaciones en que el destino de la operación original (con la que se efectúa el enlace) requiere información adicional para procesar esta operación: tal es el caso cuando se utilizan facilidades de menú (las facilidades de menú permiten a un usuario efectuar una secuencia de elecciones, cada una de las cuales depende de las anteriores).

CUADRO 1/Q.775

Usuario TC A	Usuario TC B
Petición TC-INVOCACIÓN (Prueba, Clase = 1)	Indicación TC-INVOCACIÓN (Prueba) Petición TC-RESULTADO-NL (P1)
Indicación TC-RESULTADO-NL (P1)	Petición TC-RESULTADO-NL (P2)
Indicación TC-RESULTADO-NL (P2)	Petición TC-RESULTADO-L (P3)
Indicación TC-RESULTADO-L (P3)	
Tiempo	
NOTA – El valor de la temporización es especificado por el usuario TC de origen en el momento de la invocación. Un resultado no definitivo no hace rearmar la temporización.	

Ejemplo E2: La operación consiste en la ejecución de una prueba con varias opciones. Antes de ejecutarse la prueba, se ofrecen esas opciones al originador de la misma para que efectúe una selección entre ellas (usuario TC A). Se entrelazan dos operaciones: la operación 1 es la prueba y la operación 2 es la de selección de la opción: el usuario TC A responde en primer lugar a la operación 2 antes de que el usuario TC B pueda efectuar la prueba con la opción u opciones indicadas.

Una posible secuencia de primitiva para el ejemplo 2 se indica en el Cuadro 2.

No hay ningún límite al número de invocaciones de operación que pueden enlazarse con una invocación de operación dada.

Debe observarse que, cuando una operación B está enlazada con otra operación A, éstas no necesitan estar anidadas (nested). La única condición es que la invocación de B tenga lugar antes de que se notifique la consecuencia de A. Sin embargo, la operación B no tiene por qué terminar con anterioridad a la operación A.

2.3 Facilidades relacionadas con los componentes ofrecidas a los usuarios TC

2.3.1 Invocación

Hasta ahora, se han considerado las operaciones desde el punto de vista estático. La invocación introduce un aspecto dinámico: debe diferenciarse la invocación específica de una operación de otras posibles invocaciones concurrentes de la misma operación o de otra cualquiera.

Cada activación particular de una operación se identifica mediante un ID de invocación. Este ID de invocación debe ser inequívoco. Es seleccionado por el usuario TC que origina la invocación de la operación y es transferido al usuario TC de destino que lo reflejará en su respuesta (o en cada segmento de respuesta) o en una invocación conexa: en consecuencia, correlaciona la respuesta a una invocación (o cada segmento de una respuesta) o una invocación conexa con la propia invocación.

El usuario TC tiene libertad para asignar cualquier valor al ID de invocación (índice, dirección, ...), siempre que su valor pueda hacerse corresponder a un entero que pueda codificarse en un octeto de acuerdo con las reglas de codificación especificadas en la Recomendación Q.773. Obsérvese que dicho entero toma valores entre -128 y 127.

El ID de invocación asociado con una invocación puede reutilizarse cuando se ha recibido el último o único segmento de un resultado, o cuando la TCAP indique situaciones anormales. Sin embargo, no debe reatribuirse el valor inmediatamente para otra activación de operación ya que la reatribución inmediata impediría el tratamiento correcto de algunas situaciones (véase más adelante).

Se denomina periodo de congelación al periodo durante el cual un ID de invocación está liberado pero no puede ser reatribuido. Este periodo depende de la realización.

CUADRO 2/Q.775

Usuario TC A	Usuario TC B
Petición TC-INVOCACIÓN (Prueba, Clase = 1)	Indicación TC-INVOCACIÓN (Prueba) Comienzo de la operación 1 Petición TC-RESULTADO-NL (Selección de opción, Clase = 1) Comienzo de la operación 2
Indicación TC-INVOCACIÓN (Selección de opción) Petición TC-RESULTADO-L (Opciones)	Indicación TC-RESULTADO-L (Opciones) Fin de la operación 2 Petición TC-RESULTADO-L (Resultado de la prueba)
Indicación TC-RESULTADO-L (Resultado de la prueba)	Fin de la operación 1
Tiempo	

Como los ID de invocación reciben su valor dinámicamente en el momento en que se invoca la operación, dicho valor no puede aparecer en la especificación de los protocolos de aplicación. Se trata más bien de indicar un valor «lógico», que debe sustituirse por un valor real en el momento de la ejecución, a fin de identificar una operación en un flujo único.

Teniendo en cuenta los ID de invocación, la secuencia de primitivas para el ejemplo E2 anterior sería tal como se indica en el Cuadro 3.

CUADRO 3/Q.775

Usuario TC A	Usuario TC B
Petición TC-INVOCACIÓN (1, Prueba, Clase = 1)	Indicación TC-INVOCACIÓN (1, Prueba) Petición TC-INVOCACIÓN (2, 1, Selección de opción, Clase = 1)
Indicación TC-INVOCACIÓN (2, 1, Selección de opción) Petición TC-RESULTADO-L (2, Opciones)	Indicación TC-RESULTADO-L (2, Opciones) Petición TC-RESULTADO-L (1, Resultado de la prueba)
Indicación TC-RESULTADO-L (1, Resultado de la prueba)	
Tiempo	

donde el primer parámetro de una primitiva indica un ID de invocación. Cuando deban estar presentes ambos parámetros, el segundo de ellos es el ID enlazado. Esto no es más que un convenio de notación.

2.3.2 Cancelación (por el usuario TC)

El usuario TC que pide la invocación de una operación puede parar la actividad asociada al ID de invocación correspondiente, por cualquier motivo que considere apropiado. Sin embargo, en principio, deberá reservarse la cancelación para situaciones anormales. El método normal de terminación de una operación es la recepción de un resultado o la finalización basada en la expiración de un temporizador.

La cancelación tiene solamente efecto local: no impide que el usuario TC distante envíe respuestas a una operación cancelada. Cuando se reciban tales respuestas, serán rechazadas por la TCAP como se indica en el ejemplo que sigue, que representa una secuencia de primitivas para el ejemplo E1 definidas anteriormente, en el caso en que el usuario TC A cancele la prueba tras la recepción del primer segmento del resultado.

En el Cuadro 4, el segmento P2 no lo recibe el usuario TC A: la TCAP detecta una situación de rechazo (ID de invocación no activo) y por consiguiente no lo entrega al usuario TC A, por lo que en el lado A se rechaza todo intento del usuario TC B de enviar más segmentos de respuesta.

CUADRO 4/Q.775

UsuarioTC A	Usuario TC B
Petición TC-INVOCACIÓN (1, Prueba, Clase = 1)	Indicación TC-INVOCACIÓN (1, Prueba) Petición TC-RESULTADO-NL (1, P1)
Indicación TC-RESULTADO-NL (1, P1) Decisión de cancelación: Petición TC-CANCELACIÓN (1) Indicación TC-RECHAZO-L (1, Código de problema) ...	Petición TC-RESULTADO-NL (1, P2) ...
Tiempo	

2.3.3 Rechazo (por el usuario TC)

El usuario TC es el único responsable de decidir cuándo debe enviarse un componente de rechazo o devolverse una indicación de error (fallo en la realización de una operación). El usuario TC puede rechazar un componente por cualquier razón que considere adecuada, siempre que en las especificaciones de la TCAP haya un código de problema de rechazo adecuado (por ejemplo, falta del elemento de información obligatorio en una operación, error o respuesta, operación inesperada, operación desconocida, etc.) que pueda utilizar al efecto.

De forma similar, el usuario TC decide qué código de error e información de diagnóstico (especificados como parte de las especificaciones del protocolo de usuario TC y acordados por los dos usuarios TC pares con este fin preciso) ha de utilizarse cuando se envíe un componente de error.

El cometido del usuario TC se ilustra mediante el ejemplo siguiente.

El usuario TC del lado A espera, en una situación determinada, recibir la operación Y únicamente como una operación enlazada. Al recibir del lado B un componente de invocación con un código de operación referido a Y pero ningún ID enlazado, el usuario TC del lado A puede elegir entre:

- no realizar la operación y devolver un error con algún parámetro de diagnóstico previamente determinado, especificado en la especificación de aplicación de usuario TC a tal efecto;
- rechazar el componente como una «operación no reconocida».

La interpretación del diagnóstico de error devuelto o del código de problema de rechazo corresponde al usuario TC del lado B y no se describe en las Recomendaciones relativas a la TCAP.

El rechazo de una invocación de operación o de un resultado afecta a la totalidad de la operación: para esta invocación el TC no aceptará más respuestas. El rechazo de una operación enlazada no afecta a la operación conexas por lo que respecta al TC. Los usuarios TC deben describir sus reacciones a dichas situaciones anormales como parte de su guión de aplicación.

Esto se ilustra en el Cuadro 5 en la que, en el ejemplo E2, el usuario TC A no espera el proceso de selección de opción (puede tratarse de una característica facultativa) y rechaza la operación con el código de problema «operación enlazada inesperada». El usuario TC B, puede decidir entonces la ejecución de la prueba adoptando una opción por defecto.

CUADRO 5/Q.775

Usuario TC A	Usuario TC B
Peticion TC-INVOCACION (1, Prueba, Clase = 1)	Indicacion TC-INVOCACION (1, Prueba) Peticion TC-INVOCACION (2, 1, Selección de opción, Clase = 1)
Indicacion TC-INVOCACION (2, 1, Selección de opción) Peticion TC-RECHAZO-U (2, Código de problema)	
	Indicacion TC-RECHAZO-U (2, Código de problema) Peticion TC-RESULTADO-L (1, Resultado de la prueba)
Indicacion TC-RESULTADO-L (1, Resultado de la prueba)	
Tiempo	

Cuando se rechaza una invocación de operación, el usuario TC puede decidir una nueva invocación (por ejemplo, en caso de corrupción del componente de invocación); esto podría ser una nueva invocación (nuevo ID de invocación). El usuario puede tomar también la decisión de abortar el diálogo. Un diálogo muy simple (una pregunta y una respuesta) puede no definir ningún tipo de mecanismo de recuperación, salvo cuando se trate de una operación de importancia esencial (por ejemplo, actualización de una base de datos).

2.3.4 Cancelación distante (por el usuario TC)

TC no proporciona ningún servicio específico para cancelar la ejecución a distancia de una operación en curso. El servicio de cancelación proporcionado por la primitiva petición TC-U-CANCELACION tiene únicamente un efecto local (véase 2.3.2).

Sin embargo, puede definirse un procedimiento de cancelación a distancia a nivel de usuario TC, utilizando los servicios TC existentes. Una solución para el usuario TC es incluir en uno de los ASE utilizados por el contexto de aplicación una operación cuyo objetivo sea cancelar las invocaciones existentes.

El siguiente módulo ASN.1 proporciona una descripción de dicho tipo de operación. Este tipo (y el tipo de error asociado) puede importarse en uno de los módulos utilizados por el usuario TC, de tal forma que éste pueda adjudicar los códigos de operación y error adecuados. De forma alternativa, el usuario TC puede diseñar también su propia operación de acuerdo con los mismos principios.

TCAP-Tools { ccitt recommendation q 775 modules(2) tools(1) version1(1) }

DEFINITIONS ::=

BEGIN

EXPORTS Cancel, CancelFailed, Cancelled;

IMPORTS OPERATION, ERROR, InvokeldType

FROM TCAPMessages { ccitt recommendation q 773 modules(2) messages(1) version2(2) };

Cancel ::= OPERATION

ARGUMENT InvokeldType

-- un usuario TC puede redefinir este tipo para incluir un resultado vacío de modo que pasa a ser

-- una operación de clase 1.

ERRORS { CancelFailed }

-- temporizador = 15 s

```
CancelFailed ::= ERROR
PARAMETER SET {
  problem [0] CancelProblem,
  invokeld [1] InvokeldType }
```

```
CancelProblem ::= ENUMERATED
{ unknownInvocation(0),
  tooLate (1),
  notCancellable (2) }
```

-- un usuario TC puede redefinir este tipo para incluir problemas específicos de la aplicación.

```
Cancelled ::= ERROR
```

-- un error de este tipo debiera incluirse en la lista de errores de las operaciones cancelables.

```
END
```

Es necesario incluir un error «cancelado» en la lista de errores adjunta a las operaciones cancelables. En tal caso, el usuario TC que recibe la petición de cancelación emitirá una primitiva Petición TC-U-ERROR para finalizar la operación. La recepción de un componente de error con este código de error y el correspondiente indicador TC-U-ERROR finalizará la operación en el lado de invocación.

La operación que va a cancelarse se identifica mediante el indicador invocación (invokeId) asignado al mismo en el momento de la invocación. La invocación de la operación cancelación no afecta a la máquina de estados de invocación de la operación que va a cancelarse porque el indicador invocación cursado en el argumento de la operación no es visible para la subcapa del componente.

Si falla la cancelación, se informa de un error de usuario con tres posibles diagnósticos:

- unknownInvocation (invocación desconocida): si la invocación nunca ha tenido lugar o ha sido olvidada;
- tooLate (demasiado tarde): si la invocación aún se conoce pero la ejecución se encuentra en una etapa que no permite la cancelación;
- notCancellable (no cancelable): el indicador invocación en el argumento de la operación de cancelación corresponde a una operación que no ha sido acordada por los usuarios TC como cancelable por el lado de invocación.

Cuando la cancelación se lleva a cabo con éxito y no se recibe de la operación cancelada ningún error de retorno con código de error que indique «cancelado», expirará el temporizador asociado con la operación cancelada, lo que hará que el TC emita una primitiva indicación TC-L-CANCELACIÓN.

De forma alternativa, el usuario TC que solicita la cancelación puede decidir emitir una Petición TC-U-CANCELACIÓN inmediatamente después de enviar la petición de cancelación, de tal forma que ya no exista ninguna actividad local para la operación que va a cancelarse.

La utilización del mecanismo de cancelación tiene sentido si la operación cancelación se invoca antes de que expire el temporizador asociado con la operación que va a cancelarse. Tras ello el indicador invocación puede no ser reconocido en el lado de realización, puesto que puede haberse completado la ejecución de la operación y emitido la primitiva de respuesta. La posibilidad de cancelar la ejecución de una operación tras la expiración del temporizador escapa al ámbito de estas directrices, puesto que significaría que el objetivo no es cancelar la ejecución de la operación sino las acciones subsiguientes que pueden haber sido desencadenadas en el extremo de realización por la invocación.

2.4 Situaciones anormales relacionadas con los componentes

2.4.1 Pérdida de un componente

La TCAP presupone una probabilidad de pérdida de mensajes en la red muy pequeña. Si esta probabilidad es excesiva para una aplicación, deberá utilizarse el método de servicio de red con conexión. Si algunas informaciones de protocolo requieren una calidad de servicio mejorada (por ejemplo, información de tasación) la aplicación deberá establecer sus propios mecanismos para lograr una mayor fiabilidad para esta información.

Pérdida de una invocación de operación

En el Cuadro 6 se ilustra el caso correspondiente al ejemplo E1, en que no se recibe una respuesta a la prueba antes de la expiración del temporizador.

CUADRO 6/Q.775

Usuario TC A	Usuario TC B
Petición TC-INVOCACIÓN (1, Prueba, Clase = 1)	
Límite de tiempo: Indicación TC-CANCELACIÓN-L (1)	
Tiempo	

Cuando se pierde una operación de clase 1, se informa al usuario TC de cuándo expira el temporizador asociado con la operación. Cuando se pierde una operación de clase 1 con un solo resultado, la TCAP no puede indicar si se ha perdido la invocación de operación o la respuesta. Si la aplicación debe establecer una discriminación entre estos dos casos, deberá hacerlo en el protocolo de aplicación (por ejemplo empleando una marca temporal o mediante acuse de recibo de la invocación de operación, antes de responder a la misma).

Para una operación de clase 2, la pérdida se considerará como un éxito (tanto si se perdió la invocación como el Informe de fallo). Teniendo en cuenta la probabilidad de pérdida, esto puede ser aceptable para operaciones que no sean fundamentales (por ejemplo, mediciones estadísticas).

En el caso de una operación de clase 3, el tratamiento que se da a la pérdida es el mismo que el de un fallo de operación, tanto si se ha perdido la invocación como el Informe de éxito.

Para una operación de clase 4, la pérdida no será visible por la TCAP.

Pérdida de un resultado

- La TCAP no detecta nunca la pérdida de un resultado no final.
- La pérdida de un resultado final se indicará finalmente al usuario TC cuando se alcance el límite de tiempo, pero no puede siempre interpretarse de forma inequívoca como pérdida de respuesta. Si no se ha recibido ningún resultado no final, ello puede deberse a que se haya perdido la invocación.

Pérdida de una operación enlazada

La pérdida de una operación enlazada surte el mismo efecto que la pérdida de una operación no enlazada. No tiene ningún efecto sobre la operación con la que se realiza el enlace.

Pérdida de un componente de rechazo

Este caso tendrá una frecuencia de aparición muy reducida por lo que ninguna aplicación intentará recuperarse de tal situación. Si la pérdida del rechazo afecta a una invocación de operación, cuando expire la temporización de la operación, el usuario TC que invocó la operación estimará que dicha invocación (o la respuesta) se ha perdido y reaccionará en consecuencia. Si se trata de una respuesta, el originador de la respuesta no sabrá que era incorrecta y la detección de la pérdida corresponderá al originador de la operación.

2.4.2 Duplicación de un componente

Como las duplicaciones de mensajes son muy raras en la red del sistema de señalización N° 7, no es necesario que los guiones para aplicaciones del SS N° 7 definan casos complejos en previsión de estas situaciones. Sin embargo, toda aplicación en que la duplicación fuera inaceptable, debería definir su propio mecanismo de detección de duplicación o utilizar un servicio con conexión.

Invocación de operación duplicada

Cuando una invocación de operación es duplicada (por el proveedor del servicio), el usuario TC de destino (B) puede o no detectar la operación:

- El usuario TC B detecta la duplicación: el duplicado puede rechazarse utilizando el código de problema «ID de invocación de duplicado». En este caso, el rechazo puede ser interpretado por el usuario TC distante como un rechazo de la invocación original.
- El usuario TC B no detecta la duplicación: esto puede ocurrir cuando existe una relación de amo-esclavo entre A y B, y B ejecuta la operación con desconocimiento del contexto.

Suponiendo, en el ejemplo E1, que se trata del segundo caso, una secuencia posible podría ser la indicada en el Cuadro 7.

CUADRO 7/Q.775

Usuario TC A	Usuario TC B
Petición TC-INVOCACIÓN (1, Prueba, Classe = 1)	Indicación TC-INVOCACIÓN (1, Prueba) Indicación TC-INVOCACIÓN (1, Prueba) Duplicación de la invocación no detectada Petición TC-RESULTADO-NL (1, P1) Petición TC-RESULTADO-NL (1, P1)
Indicación TC-RESULTADO-NL (1, P1) Indicación TC-RESULTADO-NL (1, P1) A detecta una situación anormal y la rechaza: Petición TC-RECHAZO-U (1, Código de problema) TC detecta una situación anormal y rechaza P2: Indicación TC-RECHAZO-L (1, Código de problema)	Petición TC-RESULTADO-NL (1, P2) Indicación TC-RECHAZO-U (1, Código de problema)
	Indicación TC-RECHAZO-R (1, Código de problema)
Tiempo	

En esta secuencia, el usuario TC B considera dos invocaciones de prueba independientes y responde a cada una de ellas. Se acepta el primer resultado P1. El usuario TC A detecta que P1 se ha recibido por segunda vez y lo rechaza. Esto concluye la operación y origina el rechazo del resultado P2 cuando se reciba (rechazado por la TCAP). En consecuencia, en el lado B ambas actividades concluirán con la recepción de los rechazos.

Duplicación de un resultado no final

La TCAP no puede detectar la duplicación de un resultado no final (RR-NL), por lo que lo entregará dos veces al usuario TC. La detección de esta situación se deja para la aplicación.

Duplicación de un resultado final

En caso de duplicación de un resultado final (RR-L), la TCAP puede detectar la situación. El segundo resultado final será considerado como anormal (la operación ha finalizado con el primer resultado «final») y la TCAP la rechaza.

En el Cuadro 8 se indica una secuencia correspondiente al ejemplo E1 en el que el tercer segmento del resultado es duplicado (por la red):

Comentario: Podría parecer una característica interesante que la TCAP descartase los duplicados en todos los casos. Sin embargo, debe observarse que:

- 1) esto exigiría otro grado de complejidad en la TCAP, que contradice las características básicas de la TCAP en el método sin conexión;
- 2) esto corresponde a una situación que es sumamente rara, al menos en la red del sistema de señalización N.º 7.

Para afrontar estas situaciones cuando lo requiera una aplicación, es mejor utilizar el método de servicio de red con conexión, ya que la duplicación puede ser detectada y tratada en las capas inferiores.

CUADRO 8/Q.775

Usuario TC A	Usuario TC B
Petición TC-INVOCACIÓN (1, Prueba, Clase = 1) Indicación TC-RESULTADO-NL (1, P1) Indicación TC-RESULTADO-NL (1, P2) Indicación TC-RESULTADO-L (1, P3) TC recibe RR-L (1, P3) Duplicación de P3: Indicación TC-RECHAZO-L (1, Código de problema)	Indicación TC-INVOCACIÓN (1, Prueba) Petición TC-RESULTADO-NL (1, P1) Petición TC-RESULTADO-NL (1, P2) Petición TC-RESULTADO-L (1, P3)
	Indicación TC-RECHAZO-R (1, Código de problema)
Tiempo	

2.4.3 Secuenciación incorrecta de los componentes

El orden de los resultados segmentados no es importante para la TCAP. Si dicho orden es importante para el usuario TC, deberán definirse mecanismos apropiados en el protocolo de aplicación (por ejemplo, introduciendo un método de numeración para identificar las respuestas intermedias en un parámetro de estas respuestas, o empleando un servicio con conexión).

Debido a una secuenciación errónea, un resultado no final puede llegar después de un resultado final; en tal situación, la TCAP rechaza el resultado no final. Esto ocurre porque la recepción del resultado final obliga a la TCAP a cerrar la máquina de estados de invocación asociada con esta operación, y entonces, cuando se recibe el resultado no final demorado, no se le puede asociar con ninguna máquina activa de estados de invocación.

La secuencia ilustrada en el Cuadro 9 se indica lo que ocurre en el ejemplo E1 cuando se recibe la última parte del resultado antes que la segunda: se informa a ambos usuarios TC.

Si se recibe una invocación de operación enlazada tras el resultado final de la operación con la que se efectúa el enlace (como consecuencia de una secuenciación errónea), se rechaza la operación enlazada.

La TCAP supone una probabilidad de secuenciación errónea muy reducida. Si la red de soporte no es satisfactoria a este respecto, deberá considerarse la utilización del servicio de red con conexión.

2.4.4 Rechazo de un componente por la TCAP

Cuando la TCAP recibe un componente (invocación de operación o respuesta) que no se ha formateado correctamente o se ha recibido fuera de contexto (por ejemplo, respuesta sin invocación de operación previa), un principio general es proceder a su rechazo, lo que significa que:

- 1) TCAP forma un componente Rechazo para informar al originador del componente que falla e informa al usuario TC local del componente Rechazo en espera del envío al extremo distante. La TCAP proporciona toda la información disponible sobre la naturaleza del componente rechazado (si el diálogo no ha sido ya terminado por el usuario TC distante);
- 2) como reacción, el usuario TC local puede tomar la decisión de abortar, continuar o finalizar el diálogo. En los dos últimos casos, cuando el usuario TC notifique su decisión a la TCAP, se informará del rechazo al usuario TC par.

CUADRO 9/Q.775

Usuario TC A	Usuario TC B
Petición TC-INVOCACIÓN (1, Prueba, Clase = 1)	Indicación TC-INVOCACIÓN (1, Prueba) Petición TC-RESULTADO-NL (1, P1)
Indicación TC-RESULTADO-NL (1, P1) Indicación TC-RESULTADO-L (1, P3) Llega la PDU RR-NL (1, P2) Resultado con secuencia errónea: rechazo: ninguna máquina de estados activa Indicación TC-RECHAZO-L (1, Código de problema)	Petición TC-RESULTADO-NL (1, P2) Petición TC-RESULTADO-L (1, P3)
	Indicación TC-RECHAZO-R (1, Código de problema)
Tiempo	

En las secciones anteriores han aparecido los posibles casos de rechazo por la TCAP. Siempre que se reconozca el ID de invocación, el rechazo por la TCAP produce la terminación de la operación. Una posible recuperación por el usuario TC consistirá en una nueva invocación de la operación terminada. Cuando la componente rechazada no es identificable, se informa al usuario TC local del componente defectuoso recibido. Se incluye un valor ID de invocación NULO en el componente de rechazo en espera de envío. Sin embargo, como este componente de rechazo no puede asociarse con ninguna invocación conocida, la reacción apropiada puede ser la de abortar el diálogo.

2.4.5 Expiración del temporizador de operación

Cuando la TCAP informa al usuario TC sobre la expiración del temporizador (indicación TC-L-CANCELACIÓN), ello indica que no puede recibirse más información sobre la invocación de la operación (en especial, ningún rechazar). Si la entidad par envía todavía información en relación a esta invocación, se descartará dicha información cuando se reciba, siempre que no se haya reatribuido el ID de invocación de la operación cancelada. Se evita, generalmente la reatribución prematura de los valores de ID de invocación, ajustando correctamente los valores del temporizador y eligiendo el instante de «congelación» de un ID de invocación una vez liberado. A fin de compensar las incertidumbres del tiempo total necesario para enviar información desde un usuario TC a otro, sin tener en cuenta el caso más desfavorable absoluto (que es además, en general, el más improbable) se necesita el establecimiento de un mecanismo dependiente de la realización, que evite la reatribución prematura de los ID de invocación.

La indicación de expiración de un temporizador corresponde a una situación anormal en el caso de las operaciones de clase 1 solamente. En ese caso, el usuario TC es consciente de que se ha perdido la invocación o la respuesta. Si no surgen efectos secundarios indeseables, puede producirse otra invocación de la misma operación tras la expiración del temporizador. Esto se indica, en el ejemplo E1, mediante la secuencia ilustrada en el Cuadro 10.

Para una operación de clase 2, la expiración del temporizador indica que para esta invocación, no se ha recibido o no se aceptará ningún fallo. Se trata de una indicación definida de éxito (para la clase 2) si se supone que no hay posibilidad de pérdida de mensaje en la red. En caso de fallo, se aplica a la clase 3 una situación similar. La indicación de expiración del temporizador para una operación de clase 4 es una decisión local.

3 Diálogos

Siempre que se emita una de las primitivas de tratamiento de operación contempladas en 2, se cursa una petición a la TCAP, pero no se envía nada al usuario TC distante, hasta que se emita una primitiva de petición de transmisión. A continuación se tratará de estas primitivas y de su relación con las primitivas de tratamiento de operación.

CUADRO 10/Q.775

Usuario TC A	Usuario TC B
Petición TC-INVOCACIÓN (1, Prueba, Clase = 1)	Indicación TC-INVOCACIÓN (1, Prueba)
Expiración del temporizador: Indicación TC-CANCELACIÓN-L (1) Petición TC-INVOCACIÓN (2, Prueba, Clase = 1)	
Tiempo	

3.1 Agrupación de componentes en un mensaje

El efecto de la emisión por parte de un usuario TC de una primitiva de tratamiento de componente (a menos que dicha primitiva tenga solamente efecto local) es la creación de un componente para incluirlo en un mensaje. El mensaje no se transmite hasta que el usuario TC lo solicite.

Obsérvese que puede también generarse un componente como consecuencia de un procedimiento de rechazo de la TCAP: en este caso, dicho componente se coloca en el siguiente mensaje de diálogo, salvo si éste es abortado.

Pueden agruparse varios componentes y enviarse al extremo distante como un único mensaje para ahorrar tara de transmisión, siempre que no se exceda el tamaño máximo de un mensaje. Esto se efectúa bajo control del usuario TC, el cual especifica, de forma explícita, cuándo desea el envío de uno o de todos los componentes en espera de transmisión. Dichos componentes son aquellos para los que el usuario TC ha emitido previamente una primitiva de componente de tratamiento con el mismo ID de diálogo.

Hasta tanto la capa de SCCP del SS N.º 7 proporcione una capacidad de segmentación y reagrupación, el usuario TC debe asegurar que no se rebasa el tamaño máximo de los mensajes SS N.º 7.

El ejemplo E3 ilustrado en la Figura 11 muestra el comienzo de un diálogo con un centro de servicio de red, en el que un conmutador solicita instrucciones (operación 1) y recibe una petición para conectar la llamada a una dirección de destino dada, y una petición para enviar información (por ejemplo, anuncios o un mensaje para su presentación visual) a la parte llamante. Ambos componentes están contenidos en un solo mensaje.

TC-COMIENZO y TC-CONTINUACIÓN son primitivas de transmisión que se describen a continuación en 3.2.

Puede haber una primitiva de transmisión para cada componente (con lo que hay un máximo de un componente en cada mensaje), o menos primitivas de transmisión que componentes, lo que permite la agrupación de componentes dentro de un mensaje. Además, la información contenida en los parámetros de las primitivas de transmisión (por ejemplo, información de direccionamiento), se aplica a todos los componentes incluidos en el mensaje.

En el lado de origen, la primitiva que solicita la transmisión aparece detrás de la primitiva de tratamiento de componente. Esto indica que la transmisión de todos los componentes precedentes se efectuará inmediatamente y evita indicar los componentes específicos que han de transmitirse con una primitiva de transmisión dada, permitiendo el uso de primitivas de transmisión sin ningún componente asociado.

En el lado de destino aparece, en primer término, la primitiva que indica la recepción de componentes transmitidos: contiene información de control necesaria a la TCAP para la entrega de cada componente (si existe) del mensaje. El parámetro «último componente» indica al usuario TC cuál es el último componente del mensaje. Los componentes se entregan al usuario TC de destino en el mismo orden en que el usuario TC de origen la transfirió a la TCAP.

CUADRO 11/Q.775

Usuario TC A	Usuario TC B
Petición TC-INVOCACIÓN (1, Proporcionar-instrucciones, Clase = 1) Petición TC-COMIENZO (Parámetros de control)	
	Indicación TC-COMIENZO (Parámetros de control) Indicación TC-INVOCACIÓN (1, Proporcionar-instrucciones) Petición TC-INVOCACIÓN (2, 1, Conectar-llamada) Petición TC-RESULTADO-L (1, Enviar información) Petición TC-CONTINUACIÓN (Parámetros de control)
Indicación TC-CONTINUACIÓN (Parámetros de control) Indicación TC-INVOCACIÓN (2, 1, Conectar llamada) Indicación TC-RESULTADO-L (1, Enviar información)	
Tiempo	

3.2 Facilidades de tratamiento de diálogo

Cuando dos usuarios TC cooperan en una aplicación se necesita generalmente más de una invocación de operación. Debe identificarse el flujo resultante de componentes de forma que:

- 1) puedan identificarse las componentes relacionadas con el mismo flujo;
- 2) puedan identificarse flujos correspondientes a diversas instancias de la misma aplicación y permitirles que discurren en paralelo.

El usuario TC considera cada uno de estos flujos como un diálogo y se identifica mediante un parámetro ID de diálogo correspondiente. La facilidad de tratamiento de diálogo ofrecida para este fin es el diálogo estructurado.

Cuando sólo se necesita un mensaje para completar una aplicación distribuida, puede emplearse el mensaje unidireccional del diálogo no estructurado. El originador no espera un Informe de la consecuencia (outcome) de la operación (es decir, sólo puede invocar operaciones de clase 4), pero puede recibir un Informe de error de protocolo, si lo hay. Este Informe de error de protocolo podrá también ser transportado en un mensaje unidireccional.

3.2.1 Diálogo estructurado

3.2.1.1 Generalidades

La utilización de diálogos permite la coexistencia de varios flujos de componentes independientes entre dos usuarios TC. Se utiliza el parámetro ID de diálogo en las primitivas de tratamiento de operación y tratamiento de transmisión (diálogo) para determinar qué componente o componentes pertenecen a cada diálogo.

En los siguientes ejemplos el parámetro ID de diálogo está representado (por convenio) por el primer parámetro de estas primitivas que comienza con la letra D. Cada usuario TC posee su propia referencia para un diálogo determinado. Aquí se representan las referencias locales (las utilizadas en la interfaz). La TCAP efectúa la correspondencia de estas referencias locales con referencias de protocolo (denominadas ID de transacción) incluidas en mensajes.

Para el tratamiento de diálogos en circunstancias normales se han definido tres primitivas que indican el comienzo del diálogo (TC-COMIENZO) su continuación (TC-CONTINUACIÓN) o su finalización (TC-FINALIZACIÓN). Puede utilizarse cada una de estas primitivas para solicitar la transmisión de 0, 1 o varios componentes. Estos componentes pueden contener información relativa a una o varias operaciones.

En el Cuadro 12 se representa, para el ejemplo E2, una secuencia posible en la que el diálogo comienza con una petición de prueba y finaliza cuando se ha enviado el resultado de la prueba.

CUADRO 12/Q.775

Usuario TC A	Usuario TC B
Petición TC-INVOCACIÓN (D1, 1, Prueba, Clase = 1) Petición TC-COMIENZO (D1, Dirección)	
	Indicación TC-COMIENZO (D2, Dirección) Indicación TC-INVOCACIÓN (D2, 1, Prueba) Petición TC-INVOCACIÓN (D2, 2, 1, Selección de opción, Clase = 1) Petición TC-CONTINUACIÓN (D2)
Indicación TC-CONTINUACIÓN (D1) Indicación TC-INVOCACIÓN (D1, 2, 1, Selección de opción) Petición TC-RESULTADO-L (D1, 2, Opciones) Petición TC-CONTINUACIÓN (D1)	
	Indicación TC-CONTINUACIÓN (D2) Indicación TC-RESULTADO (D2, 2, Opciones) Petición TC-RESULTADO-L (D2, 1, Resultado de la prueba) Petición TC-FINALIZACIÓN (D2)
Indicación TC-FINALIZACIÓN (D1, Normal) Indicación TC-RESULTADO-L (D1, 1, Resultado de la prueba)	
Tiempo	
NOTA – D1 y D2 son referencias locales para el mismo diálogo y están en correspondencia con los ID de transacción que aparecen en los mensajes.	

En los mensajes de un diálogo se permite cualquier agrupación de componentes: la TCAP no comprueba, por ejemplo, que un mensaje con el que finaliza un diálogo no incluye invocaciones de operación de clase 1.

Se presupone el intercambio dúplex completo de componentes: si un usuario TC desea introducir algunas restricciones, por ejemplo el funcionamiento en un modo síncrono, definido para usuarios ROSE, deberá introducir por sí mismo los procedimientos necesarios.

3.2.1.2 Intercambio de mensajes

La transmisión de mensajes se efectúa con la calidad de servicio correspondiente a los servicios de la capa subyacente. La TCAP no proporciona mecanismos de control de flujo ni de recuperación tras error.

- La primera primitiva de tratamiento de diálogo de un diálogo deberá indicar el comienzo del mismo (TC-COMIENZO). No podrán enviarse mensajes posteriores desde el lado en que se origina el diálogo, hasta que se reciba un mensaje en sentido opuesto, que indica la continuación del diálogo.

- Si un usuario TC trata de enviar un gran número de mensajes en un breve intervalo de tiempo, ningún mecanismo de control del flujo de la TCAP podrá evitarlo.
- Como opción puede solicitarse la entrega secuencial SCCP de clase 1 indicada por el parámetro calidad de servicio. Obsérvese que esta opción puede no estar disponible de extremo a extremo, en casos de interfuncionamiento con una red que no la proporcione.

3.2.1.3 Finalización del diálogo

La TCAP no impone ninguna restricción a la capacidad de un usuario TC para solicitar la finalización de un diálogo. Por este motivo, pueden perderse mensajes si en la aplicación no se adoptan precauciones relativas a la finalización del diálogo.

En particular, si el protocolo de aplicación permite a ambos usuarios TC la emisión de primitivas TC-FINALIZACIÓN aproximadamente al mismo tiempo, y si esas primitivas desencadenan la transmisión de componentes, es probable que algunos de esos componentes (si no todos) no se entreguen a sus respectivos usuarios TC de destino.

Corresponde a la aplicación la definición, en caso de necesidad, de sus propias reglas relativas al derecho de terminar un diálogo. La TCAP no las comprobará.

Todo mensaje recibido para un diálogo terminado se descarta si dicho mensaje solicita la finalización del diálogo, y todo mensaje que no sea una finalización o un aborto provoca el aborto del diálogo en la entidad distante.

Debe observarse que un usuario TC no puede rechazar, por medio de una primitiva de petición TC-U-RECHAZO ningún componente recibido en un mensaje de FINALIZACIÓN. Si es importante para una aplicación poder rechazar cualquier componente recibido o recibir información de los rechazos del extremo distante, todos los componentes deberán ser colocados en el mensaje de COMIENZO-INICIAL o en mensajes de CONTINUACIÓN subsiguientes.

El diálogo se termina bien sea por el método de finalización preconvenida, o mediante el envío de un mensaje de FINALIZACIÓN que no contenga componentes o que contenga componentes de RECHAZO (si son aplicables).

Las diferencias entre estas tres formas de finalización de un diálogo son las siguientes.

Finalización preconvenida

Una aplicación típica es el acceso a una base de datos distribuida en la que el usuario solicitante (usuario TC A) no sabe dónde está ubicada la información que busca. El usuario TC A difunde una petición a cada ubicación en la que podría encontrarse la información solicitada y recibirá eventualmente una respuesta del usuario TC que posee esa información. La finalización preconvenida evita el envío de mensajes desde otro destino con el contenido: «No tengo esa información». Solamente el destino que responde continuará el diálogo (si así se desea); los restantes destinos finalizarán, por convenio, el diálogo localmente. El originador de las preguntas finalizará también de forma local el diálogo con los destinos que no respondan, cuando reciba la respuesta a su pregunta. Obsérvese que se trata de un convenio entre aplicaciones: La TCAP no comprueba su cumplimiento ni ello se indica en el protocolo TCAP.

En el ejemplo E4 del Cuadro 13 se ilustra esta situación con dos destinos (B1 y B2), iniciándose dos diálogos (D1, D2) y (D3, D4). La información solicitada está disponible en B1, el cual decide continuar el diálogo.

Puede también emplearse la finalización preconvenida cuando un usuario TC desea enviar información y no espera respuesta de ningún tipo.

Finalización básica

Cuando un usuario TC emite la primitiva de petición TC-FINALIZACIÓN provoca la transmisión de todos los componentes pendientes al extremo distante. La TCAP no comprueba que todas las invocaciones de operación han recibido una respuesta cuando se solicita la finalización del diálogo, ni se notifica al usuario TC de que algunas invocaciones de operación pendientes no han recibido el resultado final.

En el extremo receptor, se considera terminado el diálogo cuando se han entregado al usuario TC todos los componentes recibidos en el mensaje que indica la finalización.

Ejemplo: el diálogo finaliza cuando en la prueba del ejemplo E1, Cuadro 14 se recibe una respuesta.

CUADRO 13/Q.775

Usuario TC A	Usuario TC B1	Usuario TC B2
Petición TC-INVOCACIÓN (D1, 1, Pregunta) Petición TC-COMIENZO (D1, Dirección) Petición TC-INVOCACIÓN (D3, 1, Pregunta) Petición TC-COMIENZO (D3, Dirección)	Indicación TC-COMIENZO (D2, Dirección) Indicación TC-INVOCACIÓN (D2, 1, Pregunta) Petición TC-RESULTADO-L) (D2, 1, Respuesta) Petición TC-CONTINUACIÓN (D2) 	Indicación TC-COMIENZO (D4, Dirección) Indicación TC-INVOCACIÓN (D4, 1, Pregunta) B2 no posee la información: Petición TC-FINALIZACIÓN (D4, local)
Indicación TC-CONTINUACIÓN (D1) Indicación TC-RESULTADO-L (D1, 1, Respuesta) D1 prosigue D3 termina localmente Petición TC-FINALIZACIÓN (D3, local)		
Tiempo		

CUADRO 14/Q.775

Usuario TC A	Usuario TC B
..... Indicación TC-FINALIZACIÓN (D1) Indicación TC-RESULTADO-NL (D1, 1, P1) Indicación TC-RESULTADO-NL (D1, 1, P2) Indicación TC-RESULTADO (D1, 1, P3) Finalización del diálogo para A Petición TC-RESULTADO-NL (D2, 1, P1) Petición TC-RESULTADO-NL (D2, 1, P2) Petición TC-RESULTADO-L (D2, 1, P3) Petición TC-FINALIZACIÓN (D2, normal) Finalización del diálogo para B
Tiempo	

Aborto por el usuario TC

La facilidad de aborto permite al usuario TC detener el diálogo en cualquier momento. Un caso típico es el abandono del servicio por parte del usuario. Las diferencias principales entre estos casos y la finalización normal, son las siguientes:

- los componentes cuya transmisión está pendiente no se envían a la entidad par;
- en el momento de producirse el aborto, puede transmitirse la información de usuario TC entre pares y entregarla al usuario TC distante.

La secuencia ilustrada en el Cuadro 15 muestra un abandono por el usuario en el ejemplo E2.

CUADRO 15/Q.775

Usuario TC A	Usuario TC B
Petición TC-INVOCACIÓN (D1, 1, Prueba, Clase = 1) Petición TC-COMIENZO (D1, Dirección)	
	Indicación TC-COMIENZO (D2, Dirección) Indicación TC-INVOCACIÓN (D2, 1, Prueba) Petición TC-INVOCACIÓN (D2, 2, 1, Selección de opción, Clase = 1) Petición TC-CONTINUACIÓN (D2)
Indicación TC-CONTINUACIÓN (D1) Indicación TC-INVOCACIÓN (D1, 2, 1, Selección de opción) Abandono del usuario; Petición TC-U-ABORTO (D1, Causa)	
	Indicación TC-U-ABORTO (D2, Causa)
Tiempo	

3.2.1.4 Situaciones anormales relacionadas con el mensaje

Estas situaciones se consideran con independencia de sus efectos en la subcapa componente.

Pérdida del mensaje

La TCAP no proporciona protección contra la pérdida de mensajes. Se identifican tres casos:

- 1) Con el mensaje comienza un nuevo diálogo: el diálogo existirá en el lado de origen solamente, no permitiéndose ningún mensaje en ningún sentido. Oportunamente, un mecanismo dependiente de la realización en el extremo de origen finaliza el diálogo.
- 2) El mensaje es la continuación de un diálogo existente: no se detecta la pérdida. La TCAP reaccionará (o no) a la pérdida de los componentes incluidos, según se indica en 2.4.1 anterior.
- 3) Con el mensaje finaliza un diálogo: la TCAP reaccionará oportunamente (mediante la expiración de un temporizador, como se describen en 2.4.1) si este mensaje contiene una respuesta a una operación de clase 1. En otro caso, un mecanismo dependiente de la realización podrá finalizar el diálogo en el extremo de destino.

Duplicación de mensajes

La duplicación de un mensaje COMIENZO hace que se abran dos transacciones, como se indica más adelante: cada una de estas transacciones tiene su propio ID local y el mismo ID de destino. El usuario TC detecta a su debido tiempo que algo es incorrecto, por lo que se abortan ambos diálogos.

La secuencia ilustrada en el Cuadro 16 muestra una duplicación del mensaje COMIENZO en el ejemplo E2:

CUADRO 16/Q.775

Usuario TC A	Usuario TC B
Peticion TC-INVOCACION (D1, 1, Prueba, Clase = 1) Peticion TC-COMIENZO (D1, Dirección)	
	Indicacion TC-COMIENZO (D2, Dirección) Indicacion TC-INVOCACION (D2, 1, Prueba) COMIENZO duplicado: Indicacion TC-COMIENZO (D3, Dirección) Indicacion TC-INVOCACION (D3, 1, Prueba) Respuesta al primer comienzo peticion TC-INVOCACION (D2, 2, 1, Selección de opción, Clase = 1) Peticion TC-CONTINUACION (D2) Respuesta al segundo comienzo Peticion TC-INVOCACION (D3, 2,1, Selección de opción, Clase = 1) Peticion TC-CONTINUACION (D3)
Indicacion TC-CONTINUACION (D1) Indicacion TC-INVOCACION (D1, 2, 1, Selección de opción)	
Indicacion TC-CONTINUACION (D1) Indicacion TC-INVOCACION (D1, 2, 1, Selección de opción) El usuario TC considera que esta invocacion es anormal y puede rechazarla o abortar uno de los diálogos:: Peticion TC-U-ABORTO (D1, Causa)	
	Indicacion TC-U-ABORTO (D3, Causa)
Tiempo	

En ese momento, existe todavía un diálogo (con ID local D2) en el lado del usuario TC B pero no en el lado del usuario TC A. El usuario TC B recibirá una indicación de la TCAP cuando expire la temporización de la operación 2 del diálogo D2 sin respuesta (indicación TC-L-CANCELACION) y puede entonces decidir abortar D2. Obsérvese que la situación sería más difícil de detectar si el usuario TC B no invocase una operación de clase 1.

La duplicación de un mensaje CONTINUACION no detectada por la TCAP

Cuando se duplica un mensaje de FINALIZACION, el segundo mensaje se recibe con un ID que no corresponde a un diálogo activo: la TCAP reacciona descartando el mensaje duplicado.

Secuenciación incorrecta de mensajes

Cuando los mensajes incorrectamente secuenciados no afectan ni al principio ni al final de un diálogo, la secuenciación incorrecta no es detectada por la TCAP, lo cual puede provocar una secuenciación incorrecta de componentes, a la que la TCAP reaccionaría como se indica en 2.4.3 anterior.

Cuando un mensaje, que indica la continuación de diálogo, llega tras un mensaje que indica el final del mismo diálogo, no será entregado y provocará que la TCAP aborte el diálogo. El usuario TC detectará probablemente la pérdida al recibir una indicación prematura de finalización del diálogo. Si la aplicación necesita recuperarse de este caso, deberá iniciarse un nuevo diálogo.

Adulteración de mensajes

Al recibir un mensaje adulterado, la TCAP reacciona como se indica en la Recomendación Q.774.

En el Cuadro 17 se indica la secuencia de primitivas cuando la TCAP decide abortar el diálogo tras recibir un mensaje adulterado en el ejemplo E2.

CUADRO 17/Q.775

Usuario TC A	Usuario TC B
Petición TC-INVOCACIÓN (D1, 1, Prueba, Clase = 1) Petición TC-COMIENZO (D1, Dirección)	
	Indicación TC-COMIENZO (D2, Dirección) Indicación TC-INVOCACIÓN (D2, 1, Prueba) Petición TC-INVOCACIÓN (D2, 2, 1, Selección de opción, Clase = 1) Petición TC-CONTINUACIÓN (D2)
Message adulterado: Indicación TC-P-ABORTO (D1, Causa)	Indicación TC-P-ABORTO (D2, Causa)
Tiempo	

3.2.1.5 Relaciones entre el tratamiento del diálogo y los componentes

A continuación se dan orientaciones sobre cuándo puede solicitarse la finalización del diálogo. Si no se respetan, la TCAP no rechazará la petición de finalización del diálogo.

Se han examinado anteriormente los problemas que pueden surgir por colisión de mensajes que soliciten la finalización del diálogo.

No se solicitará la finalización normal cuando:

- existan, para el diálogo, invocaciones de operaciones pendientes;
- el protocolo de aplicación anticipe que podrían rechazarse las respuestas transmitidas con la petición de terminación.

Muchas aplicaciones podrían no definir escenarios de recuperación como reacción a una respuesta rechazada. Esto justifica la transmisión de respuestas o de operaciones de clase 4 en un mensaje que indica la finalización de un diálogo. Las demás aplicaciones deberán usar el servicio de red con conexión o finalizar el diálogo con un mensaje que no contenga componentes y que se enviará solamente cuando ya no pueda recibirse una indicación de rechazo.

Se recomienda no enviar en un mensaje FINALIZACIÓN una operación para la que se espera una respuesta (es decir, se recomienda que el usuario TC no emita una primitiva Petición TC-FINALIZACIÓN para activar el componente asociado). Cabe señalar que esto no constituye un requisito del protocolo TCAP ni del usuario TC, sino simplemente una orientación para este último. Si el usuario TC elige enviar un componente de invocación para una operación de clase 1, 2 ó 3 en un mensaje de FINALIZACIÓN, ello no tiene repercusiones perturbadoras para las máquinas de protocolo TCAP pares o los usuarios TC. La forma en que el lado de realización descarta los componentes generados como resultado de la ejecución de la operación cuando no hay ningún diálogo activo es un asunto local. Evidentemente, el hecho de que el usuario TC que invoca la operación haya decidido incluirla en un mensaje FINALIZACIÓN significa que en este caso ha estimado conveniente invalidar la semántica inherente a la operación y que no le interesa la información sobre si la operación ha podido llevarse a cabo o no.

3.2.1.6 Aspectos de direccionamiento

El usuario TC que inicia un diálogo debe proporcionar al TC la dirección de destino y la dirección de origen. El usuario TC que proporciona la dirección de origen es responsable de presentar dicha dirección de tal forma que el TC distante pueda utilizarla, sin comprobaciones, para alcanzarlo.

Los TC por encima de SCCP sin conexión utilizan cualquiera de las opciones de direccionamiento suministradas por la SCCP. Por lo tanto, se permite cualquier combinación de tipos de dirección de destino y de origen.

Durante el establecimiento de un diálogo, la combinación de direcciones puede ser optimizada tanto por el usuario TC del lado B como por la TCAP.

El parámetro opcional de dirección de origen de la primera primitiva petición TC-CONTINUACIÓN puede utilizarse para cambiar la dirección que debe emplearse para encaminar los mensajes subsiguientes asociados con el diálogo. El destino real no deberá resultar afectado por esta modificación. Si hay que alcanzar un nuevo destino, se debe terminar el diálogo y comenzar uno nuevo con el nuevo destino.

A continuación se dan ejemplos de cambios de dirección que no modifican el destino real:

- La dirección inicial era un título global, y la dirección subsiguiente es un PC y SSN para el mismo destino a fin de permitir un encaminamiento óptimo (generalmente sin cruzar una frontera de red).
- Se utiliza un título global general para seleccionar una base de datos de un conjunto de bases de datos duplicadas. La base de datos que responde devuelve su propio título global específico.

3.2.1.7 Calidad de servicio

Las primitivas de petición de tratamiento de diálogo permiten al usuario TC solicitar una determinada calidad de servicio a la capa de red. Si el usuario TC no realiza ninguna petición específica, la SCCP selecciona opciones por defecto (es decir, sin opción de devolución, y sin secuenciamiento).

Si se solicita el control de secuencia, el usuario TC también es responsable de proporcionar la información que permita a la capa de red identificar un flujo de mensajes relacionados entre sí que deben entregarse en secuencia.

Se recomienda que los usuarios TC soliciten el control de secuencia cuando utilicen los servicios TC-RESULTADO-NL.

3.2.2 Diálogo no estructurado

Un mensaje unidireccional contendrá únicamente invocaciones de operación de clase 4 o informes de error de protocolo en tales invocaciones. En un mensaje unidireccional pueden transmitirse múltiples componentes, siempre que el mensaje no rebase el tamaño máximo permitido.

3.3 Facilidades de control de diálogo mejorado

3.3.1 Consideraciones generales

A medida que aumente el número de aplicaciones de señalización que utilizan TCAP, será necesario poder diferenciar entre ellas durante una instancia de comunicación en especial cuando un cierto número de aplicaciones residan en la misma ubicación en un nodo SS N.º 7.

Las funciones opcionales y el protocolo de la porción de diálogo ofrecen la posibilidad de señalar al principio del diálogo cuál es el protocolo de aplicación (entre los muchos posibles) que interviene en el intercambio subsiguiente de mensajes. La porción de diálogo opcional permite la negociación del contexto de aplicación y como una opción más, la transferencia transparente de los datos de usuario que no son componentes. Esta última puede utilizarse para, por ejemplo, datos de inicialización, versiones de protocolos de usuario, mejoras del contexto de aplicación, contraseñas, etc.

3.3.2 Utilización del contexto de aplicación

El usuario TC que inicia un diálogo puede proponer un contexto de aplicación a su par incluyendo un nombre de contexto de aplicación en la primitiva petición TC-COMIENZO. El contexto de aplicación se refiere al conjunto de ASE y a las reglas de coordinación asociadas que pueden necesitarse durante el diálogo.

Si el nombre del contexto de aplicación es aceptable, el usuario TC que responde puede optar por continuar o finalizar el diálogo de forma normal. Excepto si solicita una terminación preacordada, incluirá el mismo nombre AC en la primera (o única) primitiva de petición de tratamiento de diálogo que utilice.

Si el nombre AC no es aceptable, el usuario TC puede elegir entre:

- i) descartar los componentes recibidos y emitir una primitiva petición TC-U-ABORTO para indicar que rehusa el diálogo. En esta primitiva incluirá un nombre de contexto de aplicación, ya sea el recibido u otro que deberá ser utilizado por el iniciador del diálogo para realizar una nueva tentativa. TC no proporciona una característica normalizada para permitir al usuario TC proponer más de un nombre de AC alternativo. Sin embargo, tal procedimiento puede definirse fuera del ámbito del TC mediante el parámetro de información de usuario (véase 3.3.3);
- ii) continuar el diálogo pero indicar que hace uso de un AC alternativo (por ejemplo, uno que no utilice el o los ASE que no admite) incluyendo otro nombre de contexto de aplicación en la primera primitiva petición TC-CONTINUACIÓN. Los componentes recibidos pueden descartarse o no, según lo acordado previamente entre los usuarios TC;
- iii) finalizar el diálogo de forma normal indicando que la respuesta o respuestas incluidas en el mensaje FINALIZACIÓN se basan en un AC alternativo (por ejemplo, uno que no utilice el o los ASE que no admite), mediante la inclusión de otro nombre de contexto de aplicación en la primitiva petición TC-FINALIZACIÓN.

El usuario TC puede proporcionar igualmente un nombre de contexto de aplicación cuando utilice un servicio TC-UNI. En tal caso, el nombre AC indica al usuario TC par cómo debe interpretar los componentes recibidos.

Es importante señalar que la información de contexto de aplicación cursada en las APDU de tratamiento de diálogo es un nombre del tipo IDENTIFICADOR DE OBJETO. Tal nombre es una referencia a una especificación (documento) en la cual figura la descripción del contexto de aplicación. Dicho documento puede hacer referencia a otras especificaciones en las que, por ejemplo, se indique la sintaxis abstracta de algún protocolo de aplicación. Estas especificaciones pueden presentarse en una notación formal o semiformal o en lenguaje claro.

La especificación de los contextos de aplicación, sus semánticas, la asignación de un valor identificador de objeto y la distribución de esta información a todas las partes que desean comunicar constituyen el proceso de registro de un contexto de aplicación. En el caso en que los contextos de aplicación se registran como parte de las Recomendaciones sobre señalización RDSI, un ejemplo de un valor típico puede ser {ccitt recommendation q xxx ac-name(y)}, para el contexto de aplicación y-ésimo descrito en la Recomendación q.xxx.

Si bien en principio es posible una descripción de contexto de aplicación muy detallada y añadir nuevos contextos de aplicación para abarcar cualquier situación que pueda aparecer, puede resultar más sencillo mantener tales especificaciones si el número de contextos de aplicación se mantiene dentro de límites razonables. Por ejemplo, supóngase que un nombre de contexto de aplicación se refiere a la utilización combinada de los ASE A y B. En algunas circunstancias, puede ser necesario señalar que sólo se va a utilizar un subconjunto de las capacidades de A y/o B en una determinada instancia de comunicaciones. En vez de registrar un nuevo nombre de contexto de aplicación para contemplar este caso, puede cursarse la misma información en alguna sintaxis mutuamente aceptable en el campo de información de usuario de las APDU, de AARQ y AARE.

3.3.3 Utilización de los datos de usuario

Las primitivas de tratamiento de diálogo TC permiten al usuario TC solicitar al TC que transfiera información no relacionada con las facilidades de tratamiento del componente (es decir, no basadas en el paradigma de operación a distancia). Esta información se cursa en el campo de información de usuario de las PDU de control de diálogo o directamente en la porción de diálogo, una vez establecido éste.

Una situación típica en que se necesita esta facilidad es en el establecimiento de un diálogo para enviar algunos datos de inicialización a los pares (perfeccionamiento del contexto de aplicación, datos de autenticación, identificación de un subproceso de destino en el usuario TC, etc.).

Además, esta facilidad puede utilizarse para la negociación del contexto de aplicación: cuando el usuario TC rehusa un diálogo (aborto de usuario en el estado pendiente de diálogo con razón del aborto = contexto de aplicación no soportado), puede insertar una lista de nombres de contexto de aplicación alternativos en el campo de datos de usuario de

la primitiva Petición TC-U-ABORTO. Estos nombres se cursan como parte de los datos de usuario de la unidad de datos de protocolo de diálogo (ABRT). El usuario TC que ha originado la petición de establecimiento de diálogo puede efectuar una nueva tentativa con uno de esos contextos.

Para utilizar esta facilidad, los dos usuarios TC deberán definir la sintaxis y la semántica de la información a cursar, caso de existir, en cada APDU de diálogo para cada contexto de aplicación. Como el tipo ASN.1 de esta información de usuario es EXTERNO, la sintaxis de esta información puede escribirse en ASN.1 o cualquier otra notación específica de usuario. La forma en que se codifica esta información puede ser también específica del usuario. El tipo EXTERNO permite incluir un valor de datos de una Sintaxis Abstracta (en este caso una sintaxis específica de usuario) dentro de otra (la de las APDU de diálogo).

La Recomendación X.208 define el tipo EXTERNO de la forma siguiente:

```
EXTERNAL ::= [UNIVERSAL 8] IMPLICIT SEQUENCE {  
direct-reference                               OBJECT IDENTIFIER OPTIONAL,  
indirect-reference                           INTEGER OPTIONAL,  
data-value-descriptor                       ObjectDescriptor OPTIONAL,  
encoding                                     CHOICE {  
    single-ASN1-type                       [0] ANY,  
    octet-aligned                           [1] IMPLICIT OCTET STRING,  
    arbitrary                               [2] IMPLICIT BIT STRING }}
```

De las tres formas de referencia para identificar el tipo y la codificación del valor de datos contenido en la construcción EXTERNA, los usuarios TC deben utilizar la referencia directa. El nombre de referencia directa proporcionará la clave para identificar la sintaxis abstracta del valor de datos y las reglas de codificación aplicables al mismo. La referencia indirecta se utiliza para identificar el contexto de presentación, cuya utilización no está prevista actualmente en el sistema de señalización N.º 7. Además de la referencia directa, el usuario TC puede proporcionar igualmente una descripción explícita del valor de datos en una notación informal mediante el empleo del descriptor de valor de datos.

Si el valor de datos externo es un solo tipo ASN.1 y se utilizan las reglas de codificación básicas para codificar este valor, puede utilizarse cualquiera de las posibilidades previstas para el campo «codificación». Si la codificación acordada para este valor de datos da lugar a un número completo de octetos, puede utilizarse la posibilidad de codificación con «alineación de octetos» o «arbitraria». Si la codificación acordada para este valor de datos externo no da lugar a un número completo de octetos, debe utilizarse codificación «arbitraria».

Como el protocolo permite la presencia de una SECUENCIA DE EXTERNO en el campo de información de usuario opcional de las APDU de control de diálogo, los dos usuarios TC no están restringidos, cuando definan un contexto de aplicación, a ningún número en la secuencia. (Si la SCCP no proporciona segmentación, los usuarios TC tendrán que asegurarse de que no se viola la restricción en cuanto al tamaño del mensaje en el sistema de señalización N.º 7.)

3.3.4 Aspectos de retrocompatibilidad

Las nuevas funciones y protocolos descritos en los puntos anteriores son optativos, y la especificación de sus procedimientos de protocolo en las Recomendaciones Q.771 a Q.774 son fácilmente distinguibles y pueden suprimirse sin dificultades en los documentos de adquisición, las especificaciones de realización y las especificaciones de interfaz entre redes basadas en estas Recomendaciones. En ese caso, quedan los mensajes TCAP definidos en las Recomendaciones de 1988. Ninguna red está obligada a admitir estas prestaciones si no ofrece servicios que exigen estas capacidades.

Un nodo que soporte la versión 1988 de TCAP no entenderá las APDU asociadas con el contexto de aplicación generado por un nodo conforme a la presente Recomendación (1992) y, por consiguiente, abortará la transacción utilizando la causa P-ABORTO «parte de transacción incorrecta». Si un usuario TC situado en un nodo que soporta una TCAP conforme a esta Recomendación (1992) recibe un mensaje de aborto con la causa antes mencionada en respuesta a una iniciación de diálogo basada en una información de contexto de aplicación, debe interpretarlo, al menos en las situaciones en que la red soporta una combinación de realizaciones de TCAP basadas en esta Recomendación (1992) y en las Recomendaciones de 1988, como el resultado de comunicar con un nodo que soporta únicamente las Recomendaciones sobre TCAP de 1988, y no como el resultado de un error de sintaxis verdadero (lo que es algo sumamente improbable). Por consiguiente, el usuario TC puede intentar una retransmisión del mensaje sin la información de contexto de aplicación adicional, siempre que, evidentemente, esta información no sea crucial para la aplicación.

Es probable que durante un periodo de tiempo una red soporte tanto realizaciones de TCAP conformes a las Recomendaciones de 1988 como realizaciones conformes a esta Recomendación (1992), así como aplicaciones que exijan o no el mecanismo de contexto de aplicación. El despliegue de tales capacidades cae fuera del ámbito de las normas, pero convendrá tenerlo en cuenta cuando se empiecen a prestar los servicios a fin de evitar la ineficacia que supone transmitir dos veces el mensaje inicial.

4 Directrices para la formulación de especificaciones de protocolos de usuarios TC

4.1 Introducción

La Recomendación Q.1400 describe cómo se estructuran los elementos del servicio de aplicación (ASE), los contextos de aplicación (AC, *application-contexts*) y las entidades de aplicación (AE) y cómo se direcciona una AE en el sistema de señalización N.º 7. En este punto se explica esa arquitectura a partir de la descomposición funcional de una aplicación y se describe cómo deben definirse AE, AC, ASE, las operaciones y los errores.

4.2 Descomposición de la funcionalidad

Un proceso de aplicación (AP, *application process*) de señalización se comunica a través de una parte de su soporte lógico dedicada exclusivamente a comunicaciones, denominada entidad de aplicación (AE, *application entity*). Por consiguiente, una AE contiene todas las funciones necesarias para las comunicaciones entre AP distribuidos. En muchas ocasiones se observa que las funciones de comunicaciones de una cierta variedad de aplicaciones pueden agruparse en conjuntos de acciones integrados, de tal forma que cada uno de estos conjuntos puede emplearse en más de una AE. Un conjunto de acciones integrado que puede utilizarse en varias AE recibe el nombre de elemento del servicio de aplicación (ASE, *application service element*). Evidentemente, siempre hay algunas funciones de comunicaciones que son específicas de la aplicación y que pueden utilizarse únicamente para satisfacer las necesidades de comunicaciones de la aplicación para la que han sido definidas. En la Recomendación Q.1400 se examinan con más detalle los aspectos relativos a AP, AE y ASE.

Por esta razón, la TCAP puede considerarse como un ASE puesto que proporciona un medio genérico para que todas las aplicaciones de señalización comuniquen con arreglo al paradigma de operaciones a distancia por conducto de un servicio de red sin conexión. Un ASE de usuario TC es un conjunto de definiciones de operaciones a distancia que proporcionan de forma colectiva algunos protocolos de comunicaciones globales para una aplicación de señalización. La definición del ASE especifica también cuál es el usuario TC par que puede invocar cada una de las operaciones, y en qué orden. Si cualquier usuario TC puede invocar cualquiera de las operaciones, se dice que el ASE es simétrico. En 4.5 se describe la forma de definir y agrupar las operaciones.

Desde la perspectiva del usuario TC, el mecanismo para obtener los servicios de un ASE de usuario TC es la invocación de las operaciones de este ASE. Cada operación proporciona una parte del servicio del ASE de una forma intrínsecamente asimétrica, ya que es invocada por un usuario TC y ejecutada por otro (el usuario par distante). Sin embargo, los usuarios TC no son siempre asimétricos (es decir, uno limitado siempre a llevar a cabo las operaciones y el otro a invocarlas), sino que cada uno de ellos puede ser capaz de invocar o realizar una misma operación o distintas operaciones. (De hecho, la interfaz de servicio, desde el punto de vista del usuario TC (que es objeto de estudios ulteriores), puede tener un aspecto muy distinto del proporcionado por los TC. Por ejemplo, la invocación de una operación de clase 1 puede ser considerada por el usuario TC como la invocación de un servicio confirmado, mientras que, desde la perspectiva de la interfaz del servicio TC, es la consecuencia de dos servicios no confirmados, a saber, TC-INVOCACIÓN y TC-RESULTADO.)

4.3 Especificación de una entidad de aplicación y un contexto de aplicación

Un tipo de AE representa un conjunto de ASE. Cuando se comunican dos AE, las interacciones entre ellas así como las interacciones entre los ASE dentro de una AE, están determinadas por las reglas de un contexto de aplicación.

Uno o más contextos de aplicación pueden estar asociados con un tipo de AE. Cada uno de ellos representa la utilización de distintos subconjuntos del conjunto de ASE incluidos en el tipo de AE y/o de distintas reglas de coordinación.

El diseñador de la aplicación debe establecer una definición para cada tipo de AE, que debe contener al menos:

- una descripción general de los servicios sustentados por la combinación de las dos AE pares que comunican mediante un diálogo;
- una descripción general de la estructura de la AE;
- la lista de los AC soportados.

El diseñador de la aplicación debe establecer una definición de cada AC sustentado por el tipo de AE, que debe contener al menos:

- una descripción general que complete la descripción del tipo de AE;
- una definición del protocolo de aplicación completo entre las AE pares mediante:
 - i) la identificación de cada ASE utilizado por el AC (entre las que constituyen el tipo de AE), indicando cuál de las AE pares inicia el servicio;

- ii) la indicación de las reglas de coordinación entre estos ASE (por ejemplo, concatenación de las PDU procedentes de distintos ASE de usuario TC, cualquier limitación en el orden en el cual pueden invocarse las operaciones de los distintos ASE de usuario TC, etc.) además de las reglas inherentes a las especificaciones de ASE;
 - iii) la indicación de las sintaxis abstractas requeridas por los ASE;
- toda restricción especial para asegurar que las AE pares correspondientes a versiones distintas sean compatibles.

A cada contexto de aplicación se asignará un nombre. Tal nombre es un valor del tipo IDENTIFICADOR DE OBJETO, que es cursado (si es necesario) como el valor del elemento de información nombre de contexto de aplicación en la porción de diálogo.

La especificación formal de los contextos de aplicación queda en estudio.

Nota – Las primitivas que proporcionan la interfaz de servicio normalizada para el acceso a una AE como objeto compuesto quedan en estudio.

4.4 Especificación de un ASE

La especificación de un ASE de usuario TC debe comprender al menos:

- una descripción general de la finalidad del ASE y de sus procedimientos;
- la lista de operaciones soportadas así como las reglas sobre la secuencia en que pueden invocarse las operaciones y una indicación del lado que puede invocar cada operación (o de que ambos lados pueden invocarla);
- la descripción detallada de los procedimientos;
- el modo de interfuncionamiento de las distintas versiones de protocolo;
- la descripción de las interacciones entre el ASE y los TC en términos de primitivas de servicio TC;
- diagramas SDL.

La especificación formal de los elementos de servicio de aplicación queda en estudio.

Nota – Las primitivas que proporcionan una interfaz de servicio normalizada para el acceso de ASE en las AE quedan en estudio.

4.5 Especificación de operaciones y errores

4.5.1 Consideraciones generales

El conjunto de operaciones y errores que constituye una especificación de protocolo de usuario TCAP puede describirse mediante uno o varios módulos ASN.1. El número de módulos ASN.1 que habrán de utilizarse se deja a criterio del diseñador del protocolo y se examina más adelante en 4.5.5.

La notación utilizada para definir operaciones y errores se basa en la facilidad MACRO ASN.1 definida en la Recomendación X.208. El MACRO OPERACIÓN y el MACRO ERROR son los tipos de datos asociados respectivamente a una operación y a un error.

Cada operación (error) pertenece a un tipo de operación (tipo de error) que se deriva del tipo de OPERACIÓN MACRO (ERROR MACRO).

A cada tipo de operación o tipo de error deberá darse un nombre (una referencia de tipo ASN.1, que comienza con una letra mayúscula).

A cada operación o error debe darse un nombre (un valor de referencia ASN.1 que comienza por una letra minúscula).

La Recomendación X.219 prescribe que los valores para un conjunto de operaciones y errores tiene que ser único dentro de una sintaxis abstracta. Para la TCAP, esto significa actualmente que tendrán que ser únicos dentro del alcance de un número de subsistema de un grupo de números de subsistema relacionados de un contexto de aplicación.

La definición de tipo puede combinarse con la asignación de valor o efectuarse en dos pasos, como se ilustra en el ejemplo siguiente:

- La especificación de tipo y la asignación de valor están separadas

```
OperationTypeExample1 ::= OPERATION  
ARGUMENT           ParameterType1  
RESULT             ResultType1  
ERRORS            { error1, error2 }
```

```
operationExample1 OperationTypeExample1 ::= localValue 1
```

- La especificación de tipo y la asignación de valor están combinadas

```
operationExample1 ::= OPERATION  
ARGUMENT           ParameterType1  
RESULT             ResultType1  
ERRORS            { error1, error2 }  
::= localValue 1
```

La mayor o menor conveniencia de combinar las especificaciones de tipo, y por valor, así como la de utilizar un valor global o local, se discute más adelante en 4.5.6.

4.5.2 Utilización de la notación OPERATION MACRO

4.5.2.1 Utilización del tipo de notación

Un tipo de operación queda totalmente definido como una instancia de un tipo OPERATION MACRO suplementado por un comentario ASN.1 que indica el valor de temporizador asociado.

Las subcláusulas siguientes dan orientación sobre la utilización de las diversas producciones ASN.1 que forman la descripción OPERATION MACRO.

4.5.2.1.1 Especificación del argumento de operación

La siguiente producción ASN.1 indica cómo habrá de especificarse el argumento de una operación:

```
Parameter ::= ArgKeyword NamedType | empty  
ArgKeyword ::= "PARAMETER" | "ARGUMENT"
```

Si hay que proporcionar información en la invocación de Operación, debe insertarse una de las palabras clave, PARAMETER o ARGUMENT, y tiene que ir seguida del NamedType que corresponde a la estructura de datos a proporcionar; de no ser así, la palabra clave no deberá estar presente en la Descripción de Operación.

Se admite la retrocompatibilidad de ambas palabras clave con las especificaciones de usuario TC basadas en anteriores versiones de TC. No obstante, se desaconseja el uso de la palabra clave «PARAMETER» para la definición de nuevas aplicaciones.

4.5.2.1.2 Especificación de consecuencias (outcomes) positivas

Las siguientes producciones ASN.1 indican cómo especificar una operación que informa éxito:

```
Result ::= "RESULT" ResultType | empty
```

```
ResultType ::= NamedType | empty
```

Si hay que devolver información como resultado de la ejecución de una operación con éxito, la palabra clave RESULTADO tiene que ir seguida del NamedType asociado con la estructura de datos a enviar. Si no hay que proporcionar información, sino que la clase de operación indica que se trata de una operación de Informe de éxito, la palabra clave RESULT tiene que estar presente en la descripción de operación, pero se utiliza la alternativa vacío (empty) de la producción de Resultado. Si la palabra clave RESULT no se incluye en una descripción de operación, ello indica que no informa un éxito (es decir, operación de clase 2 ó 4).

4.5.2.1.3 Errores asociados

Las siguientes producciones ASN.1 indican cómo habrán de especificarse operaciones que informen un fallo:

```
Errors ::= "ERRORS" "{" ErrorNames "}" | empty
```

```
ErrorNames ::= ErrorList | empty
```

```
ErrorList ::= Error | ErrorList "," Error
```

```
Error ::= value (ERROR) | type
```

Si la operación informa fallo, la palabra clave ERRORS deberá incluirse e ir seguida por la lista de errores asociados; en otro caso, esta palabra clave no deberá estar presente. Los errores incluidos en la lista pueden ser referenciados ya sea utilizando una referencia de tipo, o una referencia de valor (es decir, un código de error).

4.5.2.1.4 Especificación de operaciones enlazadas (linked operations)

Las siguientes producciones ASN.1 indican cómo habrán de especificarse operaciones enlazadas:

```
LinkedOperations ::= "LINKED" "{" LinkedOperationNames "}" | empty
LinkedOperationNames ::= OperationList | empty
OperationList ::= Operation | OperationList "," Operation
Operation ::= value (OPERATION) | type
```

Si la operación es la operación progenitora (parent operation) de un conjunto de operaciones enlazadas, la palabra clave LINKED debe incluirse e ir seguida por la lista de operaciones vástagos (child operations). Las operaciones vástagos incluidas en la lista pueden ser referenciadas ya sea utilizando una referencia de tipo o una referencia de valor (es decir, un código de operación).

4.5.2.2 Utilización de la notación de valor

La notación de valor para operación es o bien la notación para el valor de un elemento de tipo INTEGER o la notación para un elemento de tipo OBJECT IDENTIFIER. Esto depende de si a la operación se le asigna un valor local o un valor global.

4.5.2.3 Especificación de temporizadores

El valor de temporizador asociado con un tipo de operación tiene que indicarse como un comentario ASN.1 frente a la descripción ASN.1 MACRO del tipo de operación.

4.5.3 Utilización de la notación ERROR MACRO

La notación de tipo para un error es la palabra clave ERROR seguida opcionalmente por la palabra clave PARAMETER y el NamedType asociado a la información que puede enviarse como parámetro de error. La palabra clave PARAMETER no estará presente si no se asocia información con la condición de error.

La notación de valor para un error es o bien la notación para el valor de un elemento de tipo INTEGER o la notación para un elemento de tipo OBJECT IDENTIFIER. Esto depende de si a la operación se le asigna un valor local o un valor global.

4.5.4 Ejemplos de descripciones de operaciones y errores

Esta cláusula ilustra la parte de especificación de protocolo que trata de las definiciones de operaciones y errores asociados para un ASE de usuario TC simple. La finalidad de operaciones y errores se describe brevemente en forma textual. Seguidamente las operaciones y errores, así como los tipos de datos asociados, se describen formalmente en un módulo ASN.1.

El siguiente ejemplo se basa en un diálogo ficticio de tipo «llamada telefónica gratuita» (freephone) entre un centro de conmutación y una base de datos de tipo «llamada telefónica gratuita».

4.5.4.1 Finalidades de operaciones y errores

4.5.4.1.1 Proporcionar información de encaminamiento

Esta operación la invoca un centro de conmutación para pedir a una entidad distante que proporcione información de encaminamiento para establecer una llamada a un abonado. La información de encaminamiento proporcionada puede ser un número a que se reenvía (forwarded-to number) y puede depender del número de la parte llamante y/o del servicio básico solicitado. En este último caso se invoca la operación vástago getCallingPartyNumber.

4.5.4.1.2 Obtener número de parte llamante

Esta operación es invocada por un elemento de red para pedir a un centro de conmutación que proporcione el número de la parte llamante asociado con un establecimiento de llamada.

4.5.4.1.3 Número llamado no válido

Este error es retornado por un elemento de red para indicar que el número llamado recibido no es conforme con el esquema de numeración soportado.

4.5.4.1.4 Abonado no alcanzable

Este error es retornado por un elemento de red para indicar que en ese momento no hay información de encaminamiento disponible que corresponda al número llamado.

4.5.4.1.5 Llamada prohibida

Este error es retornado por un elemento de red para indicar que la llamada no puede establecerse porque el número llamante está en conflicto con las condiciones de prohibición asociadas al número llamado.

4.5.4.1.6 Número de parte llamante no disponible

Este error lo retorna un centro de conmutación para indicar que no se puede proporcionar el número de la parte llamante.

4.5.4.1.7 Fallo de procesamiento

Este error lo retorna un elemento de red para indicar un fallo de procesamiento.

4.5.4.2 Especificación ASN.1

El siguiente módulo ASN.1 especifica las operaciones y los errores asociados, así como los tipos de datos que corresponden a los elementos de protocolo descritos anteriormente. En este ejemplo la definición tipo de operaciones y errores se combina con la asignación del valor.

```
TCAP-Examples { ccitt recommendation q 775 modules(2) examples(1) version1(1) }
DEFINITIONS ::=
BEGIN

IMPORTS OPERATION, ERROR
FROM TCAPMessages { ccitt recommendation q 773 modules(2) messages(1) version2(2) };

provideRoutingInformation          OPERATION
ARGUMENT                          RequestArgument

RESULT                             RoutingInformation

ERRORS                              { invalidCalledNumber,
                                     subscriberNotReachable,
                                     callBarred,
                                     processingFailure }

LINKED                              { getCallingPartyAddress }
-- temporizador T-pi = 10 s
 ::= localValue 1

getCallingPartyAddress            OPERATION
RESULT                            CallingPartyAddress

ERRORS                              { callingPartyAddressNotAvailable,
                                     processingFailure }

-- temporizador T-gp = 5 s
 ::= localValue 2

invalidCalledNumber ERROR ::= localValue 1
subscriberNotReachable ERROR ::= localValue 2
calledBarred ERROR ::= localValue 3
callingPartyAddressNotAvailable ERROR ::= localValue 4
processingFailure ERROR ::= localValue 5
-- tipos de datos

RequestArgument ::= SEQUENCE {
calledNumber      IsdnNumber,
basicService      BasicServiceIndicator OPTIONAL
}

RoutingInformation ::= CHOICE {
reroutingNumber      [0] IMPLICIT IsdnNumber,
forwardedToNumber    [1] IMPLICIT IsdnNumber }

BasicServiceIndicator ::= ENUMERATED {
speech (0),
unrestrictedDigital (1) }
```

CallingPartyAddress ::= IsdnNumber

IsdnNumber ::= SEQUENCE {
typeOfAddress TypeOfAddress,
digits TelephonyString }

TypeOfAddress ::= ENUMERATED {
national (0),
international (1),
private (2) }

**TelephonyString ::= IA5String (FROM ("0"|"1"|"2"|"3"|"4"|"5"
|"6"|"7"|"8"|"9"|"*"|"#")) (SIZE (1..15))**

END

4.5.5 Utilización de módulos ASN.1

Un módulo es una construcción ASN.1 en la que el diseñador del protocolo reúne varias definiciones de tipos y valores.

Teóricamente, ASN.1 no impone restricciones en cuanto al número de módulos que puedan ser utilizados para definir un protocolo. Todas las definiciones pueden estar contenidas en uno o en muchos módulos. Sin embargo, si las definiciones contenidas en un módulo se necesitan en otro (por ejemplo cuando el error utilizado por una operación se define en otro módulo), la definición correspondiente se hace disponible EXPORTándola desde el módulo en que está definida e IMPORTándola en el módulo en el que se utiliza. Esto se aplica a todos los objetos ASN.1, independientemente de que estén definidos por tipo o por valor.

Ello confiere al diseñador de aplicaciones la libertad de estructurar los módulos de acuerdo con sus necesidades, o según las reglas convencionales que él mismo se imponga. Por ejemplo, un solo módulo podría contener todas las definiciones, particularmente en un entorno con una sola AE y un solo ASE. Otra posibilidad sería que hubiera un módulo para cada definición de ASE, conteniendo cada módulo todas las operaciones y errores utilizados exclusivamente por ese ASE. En el otro extremo, todas las operaciones y errores podrían estar definidos en un módulo de «registro central» y ser exportadas para su utilización en los otros módulos en que estén definidos los ASE.

4.5.6 Asignación y gestión de los códigos de operación y de error

4.5.6.1 Consideraciones generales

En 4.1 a 4.5.4 se describe cómo pueden especificarse AE, ASE, operaciones y errores. Se expone también cómo los ASE utilizan operaciones y errores y cómo se emplean los propios ASE para definir el contexto de aplicación entre dos AE pares.

Las AE y los ASE constituyen unos instrumentos de modelado y especificación muy útiles para el diseño de protocolos de aplicación. En última instancia, durante un diálogo entre dos usuarios TC, todo protocolo de aplicación basado en la TCAP está constituido por un intercambio de valores de datos de tipos de operaciones y de errores, identificados respectivamente por sus códigos de operación o error. El único requisito del sistema de señalización N.º 7 (y de ROSE) es que los códigos de operación y de error sean unívocos dentro de la sintaxis abstracta. Como actualmente no hay forma de señalar de manera explícita la sintaxis abstracta a la que pertenece un determinado código de error o de operación, el diseñador de la aplicación debe asegurarse de que estos códigos son únicos en el ámbito de un número de subsistema o de un contexto de aplicación. Cuando el ámbito de los códigos de operación y error sea un contexto de aplicación, el diseñador de la aplicación deberá asegurar igualmente que el nombre del contexto de aplicación se curse al extremo distante mediante el protocolo de la porción de diálogo.

Hay muchos métodos posibles por lo que respecta a la asignación y gestión de los códigos de operación y de error, y es preciso considerar muchos factores. Dos factores muy importantes son la estructura de AE/ASE y la reutilización de operaciones y errores.

4.5.6.2 Importación y exportación de operaciones y errores

Como cualquier otro tipo ASN.1, Operaciones (Operations) y errores (Errors) pueden ser exportados e importados entre módulos ASN.1. Este método puede utilizarse cuando sea necesario definir una operación cuyo tipo corresponde a una operación existente, pero el valor que deba asignarse a esta nueva operación es diferente del asignado a la operación existente (es decir, para fines de unicidad). Esto se ilustra por el siguiente ejemplo en el cual objectIdentifier1 y objectIdentifier2 son identificadores ficticios.

```

ExportingModule { objectIdentifier1 } DEFINITIONS ::=
BEGIN
EXPORTS operation1, OperationTypeA, error1, ErrorTypeA;

IMPORTS OPERATION, ERROR FROM TCAPMessages
{ ccitt recommendation q 773 modules(2) messages(1) version2(2) };

operation1          OPERATION
ARGUMENT           ParameterType1
RESULT             ResultType1
ERRORS             { error1 }
 ::= localValue 1

OperationTypeA ::= OPERATION
ARGUMENT           ParameterTypeA
RESULT             ResultTypeA
ERRORS             { ErrorTypeA }

operation2 OperationTypeA ::= localValue 2

error1 ERROR
PARAMETER DiagnosticType1
 ::= localValue 1

ErrorTypeA ::= ERROR
PARAMETER DiagnosticTypeA

error2 ErrorTypeA ::= localValue 2
-- Se señala que ParameterType1, ResultType1, ParameterTypeA, ResultTypeA,
-- DiagnosticType1 and DiagnosticTypeA han de definirse en alguna parte.
-- Si no se definen en este módulo, han de ser importados desde el módulo en el que se definen.

END

ImportingModule { objectIdentifier2 } DEFINITIONS ::=
BEGIN
IMPORTS OPERATION, ERROR FROM TCAPMessages
{ ccitt recommendation q 773 modules(2) messages(1) version2(2) };
operation1, OperationTypeA, error1, ErrorTypeA
FROM ExportingModule { objectIdentifier1 };

operation2 OPERATION
ARGUMENT ParameterTypeX -- ha de definirse en alguna parte del módulo
 ::= localValue 2

error2 ERROR ::= localValue 2
-- el valor 2 ya está utilizado. Así pues, el valor 3 se asigna a los objetos importados.
operationA OperationTypeA ::= localValue 3
errorA ErrorTypeA ::= localValue 3

END

```

4.5.6.3 Influencia de la estructura de ASE/AE en la administración de los códigos de operación y de error

Con respecto a la estructura de AE/ASE, las opciones son las siguientes:

Enfoque monolítico – Un AC, un ASE

Conceptualmente, éste es el enfoque más sencillo. El protocolo de aplicación se define por una AE que comprende un solo ASE (además de TCAP). Todas las operaciones utilizadas en ese ASE podrían estar definidas en un solo módulo ASN.1, que contuviera también las definiciones del ASE y del AC. Dentro del protocolo, todas las operaciones y errores se identifican unívocamente al asignárseles un valor (entero) local único.

La ventaja de este esquema es su sencillez, y su inconveniente radica en que no permite identificar independientemente los bloques de construcción que puedan evolucionar separadamente en el seno de la estructura de AE.

Una AE que comprende más de un ASE

Al definir un protocolo de aplicación, el diseñador puede optar por estructurar una AE (y por consiguiente los AC) de modo que comprenda dos o más ASE. Por ejemplo, puede agrupar los elementos de protocolo referentes a la autenticación de usuario en un ASE separado (que podría ser reutilizado en otro protocolo), y los concernientes a la

consulta de bases de datos propiamente dicha en otro ASE. Esto puede facilitar el diseño de un sistema modular pero, cuando se combinen todos los ASE constitutivos para formar la AE, habrá que tener cuidado de asegurarse de que a las operaciones y errores distintos contenidos en ASE diferentes no se les haya asignado el mismo valor.

La utilización de una misma operación/error en dos ASE diferentes de una misma AE no causa ningún problema. Si los valores asignados a esa operación son los mismos en cada caso, dentro del protocolo habrá solamente una operación/error asociado con ese valor. Si se asignan valores diferentes, aunque en el protocolo aparecerá como si hubiese dos operaciones/errores diferentes, en la realización de la aplicación estos dos valores distintos provocarán la invocación/identificación de la misma operación/error.

Sin embargo, si dentro de una misma AE se asigna a una operación definida en un ASE el mismo valor que tiene asignada una operación diferente en otro ASE, es evidente que esto causará un problema. Cuando un ASE se utiliza en una sola AE, un simple esquema de asignación de código puede evitar este problema. En cambio, cuando se utiliza el mismo ASE en varias AE, la situación será difícil de controlar y las únicas soluciones «seguras» serán las descritas en los apartados i) a iii) siguientes:

- i) Dos o más protocolos ASE comparten valores locales comunes de operación/error

Cuando se definen los ASE, el diseñador o diseñadores del protocolo asignan los valores de tal forma que no pueda producirse ningún conflicto. Ello requiere una coordinación en las tareas de definición del ASE, y significa que los ASE comparten la misma sintaxis abstracta.

Un inconveniente de este esquema es que si se utiliza uno de los ASE en más de un contexto (es decir, junto con un conjunto distinto de ASE), es casi imposible evitar conflictos de valores en todas las combinaciones posibles.

- ii) Asignación de valores globales (identificadores de objeto) a operaciones y errores

Como un identificador de objeto es único en todo el sistema de señalización N.º 7, no hay peligro de conflicto de valores cuando se combina un ASE con cualquier otro.

Este esquema tiene el inconveniente de que, cuando un identificador de objeto está codificado, es más largo que un entero simple.

- iii) Compartición de operaciones/errores asignando tipos cuando se definen operaciones/errores en vez de valores

Esta solución supone que la definición del tipo de las operaciones y errores es independiente de la asignación de los valores.

Cuando un diseñador del protocolo define un contexto de aplicación, recopila todos los tipos de operación y error utilizados por los ASE requeridos y los asigna valores adecuados, de forma que no se produzcan conflictos.

Haciendo esto puede considerarse que el diseñador del protocolo define un nuevo conjunto de ASE que son isomórficos con respecto a los existentes, y que difieren únicamente por los valores de sus operaciones y errores.

4.5.6.4 Reutilización de operaciones y errores

Independientemente del número de ASE incluidos en un protocolo, existen situaciones en que conviene incluir una operación o error existente cuando se define un nuevo ASE.

La operación o el error pueden reutilizarse de una de las siguientes formas:

La operación se importa en uno de los módulos que definen uno de los ASE. Esto sólo es posible si se asegura que no haya conflictos de valores.

Esto puede lograrse si:

- i) Existe un registro central de operaciones y errores que utiliza únicamente valores de una gama reservada que nunca es empleada por las operaciones específicas del ASE. Este enfoque impone una limitación en cuanto a los ASE de usuario TC, que tal vez no se satisfaga en un entorno más amplio (es decir, si van a utilizarse operaciones o errores de protocolos ISO o DSS 1).
- ii) Se adjudican valores globales a las operaciones y errores. El inconveniente de esto es que un valor global exige codificar más octetos que uno local y necesita igualmente un registro oficial en el árbol de identificador de objeto.

- iii) El tipo de operación o de error se importa en uno de los módulos que definen uno de los ASE, donde se ha asignado un valor adecuado. Ello supone que el protocolo de exportación utiliza el método de dos pasos para la definición de operaciones y errores, o que los tipos de operaciones y de errores necesarios están incluidos en un registro central.
- iv) La operación o el error están completamente redefinidos. No obstante, puede importarse parte de la definición original (por ejemplo, el tipo del argumento).

4.6 Especificaciones de tipos de datos

4.6.1 Generalidades

Como se ha indicado anteriormente, el tipo de información que puede acompañar a una invocación de operación, el informe de un éxito o el informe de un fallo, se especifica como un tipo de datos ASN.1. Esto también es válido para la información que puede intercambiarse como datos de usuario de la porción de diálogo.

Este tipo de datos puede ser un tipo incorporado (por ejemplo, tipo entero, tipo booleano, tipo nulo, tipo de cadena de octetos, etc.) o estructurado (por ejemplo, tipo de secuencia, secuencia de tipo, tipo de elección, etc.). Puede también derivarse de estos tipos mediante subtificación (por ejemplo, restricción del tamaño, gama de valores) o rotulado.

4.6.2 Utilización de rótulos

ASN.1 proporciona un mecanismo de rotulación que permite definir un tipo isomórfico con respecto a otro existente, que por ende sólo se diferencia de éste por su rótulo.

Como se indica claramente en la Recomendación X.208, los rótulos (ASN.1) están destinados a ser utilizados exclusivamente por máquinas, fundamentalmente para facilitar el proceso de decodificación.

Los rótulos no están destinados a ser utilizados para la identificación directa de los elementos de información, según éstos se perciben desde el punto de vista de un proceso de aplicación local. La forma en que se identifican localmente estos elementos de información es una cuestión de realización y depende del diseño del soporte lógico y del lenguaje utilizado para manipular la representación interna de los datos. A este respecto cabe señalar que se requieren rótulos distintos principalmente en las siguientes situaciones:

- los elementos de información son miembros de un conjunto (no ordenado, es decir, un tipo de conjunto) y por consiguiente su posición relativa no puede utilizarse para discriminar entre dos elementos de información del mismo tipo (y por consiguiente con el mismo rótulo);
- los elementos de información son miembros de un conjunto ordenado (es decir, un tipo de secuencia) pero la presencia o ausencia de elementos opcionales hace imposible discriminar entre la presencia de un elemento opcional y la presencia de un elemento de información del mismo tipo inmediatamente siguiente;
- dos apariciones de un mismo tipo base en un tipo de elección.

Hay cuatro clases de rótulos. Además de la clase UNIVERSAL que se utiliza para identificar un tipo incorporado, se definen tres clases para permitir la definición de tipos isomórficos con fines de decodificación:

- La clase *APPLICATION-WIDE* – Los rótulos asignados en esta clase pueden utilizarse para identificar directamente la estructura del tipo de datos que habrán de ser decodificados/codificados. Los rótulos asignados en esta clase son significativos a través de una aplicación y no se utilizarán cuando haya riesgo de conflicto entre valores. La clase APPLICATION-WIDE debe utilizarse solamente si la aplicación es un dominio «cerrado» o si hay un registro común (common registry).
- La clase *CONTEXT-SPECIFIC* – Los rótulos asignados en esta clase sólo son significativos en un dominio definido. Por lo tanto, el proceso de decodificación identifica la estructura de datos a decodificarse tanto a partir del valor del rótulo como del contexto en que aparece. Existe el entendimiento común de considerar el contexto limitado a la construcción inmediatamente superior.
- La clase *PRIVATE*, que es muy similar a la clase APPLICATION-WIDE pero está fuera del ámbito de la normalización.

Debe señalarse que la clase CONTEXT-SPECIFIC es la única (cuando se utiliza correctamente) que asegura que nunca habrá conflicto entre valores, cuando se efectúa importación/exportación de tipos de datos entre módulos.

4.6.3 Instancias y tipos

Es necesario distinguir claramente entre un tipo de datos de una instancia (instance) de un tipo de datos, que es la representación abstracta de un elemento de información transportado en una unidad de datos. Para facilitar la especificación y acrecentar la legibilidad, ASN.1 proporciona una notación NamedType que permite calificar una instancia específica de un tipo de datos utilizando un identificador ASN.1.

Debe observarse que no es necesario definir un tipo de datos por cada elemento de información. Cuando dos elementos de información son sintácticamente equivalentes es evidentemente más convenientemente representarlos como dos instancias del mismo tipo de datos, o, si se requiere para la decodificación, como instancias de dos tipos derivados del mismo tipo de datos por una rotulación CONTEXT-SPECIFIC y cuyas definiciones aparecerán entonces solamente dentro de la definición de la construcción superior (es decir, los rotulados sólo se definen en el contexto específico de la construcción superior).

4.6.4 Exportación e importación de elementos de información

Es posible que los protocolos de señalización basados en la TCAP tengan que utilizar elementos de información definidos en otras especificaciones de protocolo de señalización. Más bien que definir un nuevo elemento de información, deberá preferirse utilizar un mecanismo de importación. Los tipos de datos pueden ser importados formalmente o informalmente, lo que depende de la manera en que está especificado el protocolo exportante.

- El protocolo exportante se especifica utilizando un módulo ASN.1 que exporta los tipos de datos requeridos: los tipos de datos requeridos pueden ser importados formalmente en el módulo donde hay que definir el nuevo protocolo.
- El protocolo exportante no se especifica utilizando módulos ASN.1: un modo conveniente consiste en definir un nuevo tipo de datos isomórfico al tipo cadena de octetos y especificar informalmente una referencia a la especificación en que se define la estructura interna (es decir, utilizando un enunciado de comentario).

4.7 Especificación de sintaxis abstractas

Las especificaciones de ASE y AC suponen una referencia a una o varias sintaxis abstractas. Cada una de ellas representa, en un nivel abstracto (es decir, independiente de las técnicas de codificación), conjuntos de valores de datos que pueden intercambiarse durante la comunicación.

Actualmente no es necesario asignar explícitamente un nombre a la sintaxis abstracta constituida por los mensajes TC para una aplicación determinada, puesto que esta sintaxis abstracta viene indentificada implícitamente por el número del subsistema que direcciona la AE. Sin embargo, la estructura de la información de usuario cursada en la porción de diálogo se deberá definir como parte de una o varias otras sintaxis abstractas.

Por consiguiente, el diseñador de protocolo que desee informaciones de usuario que no sean componentes cursados por las TC, definirá en primer lugar una o varias sintaxis abstractas que engloben todos los tipos de datos cuyos valores puedan ser cursados.

Asignará igualmente un nombre a cada una de estas sintaxis abstractas. Dicho nombre, que es un valor del tipo IDENTIFICADOR DE OBJETO, servirá de referencia directa cuando se curse el valor real como parte de un valor construido de tipo EXTERNO, como se especifica en la Recomendación Q.773.

Actualmente no hay un método formal para especificar una sintaxis abstracta; sin embargo, cuando esta sintaxis puede describirse en ASN.1, la manera más sencilla consiste en definir un tipo de elección construido a partir de todos los tipos de datos que constituyen la sintaxis abstracta.

Por consiguiente una sintaxis abstracta puede definirse informalmente incluyendo la frase siguiente en las especificaciones del protocolo:

«El conjunto de valores de datos del tipo Module-X.Type-A forma una sintaxis abstracta identificada por el siguiente nombre de sintaxis abstracta: <objectIdentifierValue>».

En la frase anterior, Type-A es el nombre del tipo de elección, y Module-X es el nombre del módulo en el que se le define.

En este contexto, el nombre de la sintaxis abstracta se refiere también implícitamente a las reglas de codificación que deben aplicarse a la sintaxis abstracta. Tales reglas de codificación, que pueden ser las definidas en la Recomendación X.209 (pero no necesariamente), deben ser acordadas *a priori* entre los usuarios TC.

El siguiente ejemplo, que ilustra un módulo que define InitData como un conjunto de tres unidades de datos de protocolo que forman una sintaxis abstracta utilizada en el establecimiento del diálogo para transferir una lista de unidades funcionales soportadas o información de autenticación, muestra los tipos ASN.1 requeridos:

```

InitModule DEFINITIONS ::=
BEGIN

InitData ::= CHOICE {
functionalUnits [0] IMPLICIT FunctionalUnits,
authenticationInfo [1] IMPLICIT AuthenticationInfo }

FunctionalUnits ::= SEQUENCE OF FunctionalUnit
FunctionalUnit ::= ENUMERATED {
unit(1), unit2(2), unit3(3) }

AuthenticationInfo ::= SEQUENCE {
algorithm OBJECT IDENTIFIER,
signature OCTET STRING }

-- El conjunto de valores de datos del tipo type InitModule-InitData forma
-- una sintaxis abstracta definida por lo siguiente
-- abstract-syntax-name: "<objectIdentifierValue>"

END

```

4.8 Reglas de codificación

La sintaxis concreta de mensajes TCAP (es decir, el tren de bits intercambiado entre TCAP pares como datos de usuario de mensajes SCCP) se obtiene aplicando las reglas básicas de codificación a la descripción de la sintaxis abstracta de mensajes TCAP (incluidos los elementos de usuario TCAP, excepto los cursados como valor de un tipo EXTERNO, por ejemplo, el campo de información de usuario de una APDU de control de diálogo). Las reglas básicas de codificación se definen en la Recomendación X.209, y en la Recomendación Q.773 se indican algunas restricciones menores relacionadas con la codificación de la porción TCAP.

La información de usuario cursada como valor de un tipo EXTERNO puede también codificarse de acuerdo con las reglas básicas de codificación (pero no necesariamente). En ese caso, el correspondiente nombre de la sintaxis abstracta sirve también como referencia implícita a las reglas de codificación aplicadas (véase 3.3.3).

Cabe señalar que las reglas básicas de codificación presentan varias opciones, especialmente para la codificación de longitudes. Esto significa que una realización deberá poder decodificar una unidad de datos independientemente de las condiciones de codificación seleccionadas por la entidad emisora.

