



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

Addendum 1
Q.1400

(02/95)

RED INTELIGENTE

**MARCO DE ARQUITECTURA
PARA DESARROLLAR PROTOCOLOS
DE SEÑALIZACIÓN Y DE OPERACIONES,
ADMINISTRACIÓN Y MANTENIMIENTO
UTILIZANDO CONCEPTOS DE LA
INTERCONEXIÓN DE SISTEMAS ABIERTOS**

**Addendum 1 a la
Recomendación UIT-T Q.1400**

(Anteriormente «Recomendación del CCITT»)

PREFACIO

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

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

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución N.º 1 de la CMNT (Helsinki, 1 al 12 de marzo de 1993).

El addendum 1 a la Recomendación UIT-T Q.1400 ha sido preparada por la Comisión de Estudio 11 (1993-1996) del UIT-T, y fue aprobada por el procedimiento de la Resolución N.º 1 de la CMNT el 7 de febrero de 1995.

NOTA

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

© UIT 1995

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

**MARCO DE ARQUITECTURA PARA DESARROLLAR
PROTOCOLOS DE SEÑALIZACIÓN Y DE OPERACIONES,
ADMINISTRACIÓN Y MANTENIMIENTO UTILIZANDO CONCEPTOS
DE LA INTERCONEXIÓN DE SISTEMAS ABIERTOS**

(Ginebra, 1995)

Se han acordado nuevas reglas sobre la compatibilidad con respecto a anteriores y futuros protocolos especificados en ASN.1 (compatibilidad «hacia adelante» y «hacia atrás»). Estas reglas son sencillas de utilizarse y se basan en las últimas Recomendaciones de la serie X.680. Sobre ASN.1 la subcláusula 12.5/Q.1400 (1993) se sustituye por el texto siguiente:

12.5 Reglas de compatibilidad para protocolos de capa de aplicación especificados en ASN.1

Se prevé que de vez en cuando será necesario introducir pequeñas ampliaciones a un Protocolo de Aplicación. Una sintaxis abstracta se amplía cuando se amplía su tipo asociado (es decir, si se trata de un tipo selección, puede ampliarse añadiendo un nuevo componente o ampliando un componente existente). Un modo de ampliar una PDU (o cualquier tipo estructura) consiste en ampliar el tipo de cualquiera de sus componentes. Para soportar tales ampliaciones, es preciso asegurarse de que, en efecto son de carácter menor. Pueden considerarse menores las extensiones siguientes:

- adición de un elemento de información que puede potenciar una actividad pero no es esencial para realizar la actividad básica (por ejemplo, lista de opciones de encaminamiento adicionales);
- adición de un elemento de información para añadir una capacidad que no es esencial para la capacidad básica (por ejemplo, adición de «Nombre» además de «Número» a efectos de la visualización en el terminal).

En los casos citados, no es necesario definir un nuevo nombre de Contexto de Aplicación, pero el proceso de aplicación receptor debe incluir procedimientos de compatibilidad hacia adelante para tratar la información desconocida.

Pueden considerarse mayores los siguientes tipos de ampliaciones:

- adición de un nuevo procedimiento;
- cambio fundamental de un procedimiento (por ejemplo, «aplicar este procedimiento dos veces»).

En los casos mencionados, debe definirse un nuevo nombre de Contexto de Aplicación.

Las siguientes reglas de compatibilidad hacia adelante y hacia atrás se aplican a los protocolos de capa de aplicación especificados que utilizan ASN.1

12.5.1 Reglas de compatibilidad hacia atrás

Las modificaciones de todas las futuras versiones de las Recomendaciones del UIT-T sobre protocolos de señalización basados en ASN.1 a partir de 1992 se llevarán a cabo de forma que se garantice la compatibilidad con versiones anteriores.

El objetivo de las subcláusulas siguientes es proporcionar directrices sobre los tipos de cambios que se pueden hacer a una especificación de protocolo asegurando la compatibilidad con el protocolo original.

Los cambios que no afectan a la sintaxis de transferencia (es decir, los bits y bytes intercambiados entre entidades pares) o que la amplían, son compatibles. En otras palabras, la compatibilidad hacia atrás significa que una PDU codificada del protocolo original es una PDU codificada válida para el nuevo protocolo.

Aparte de estos casos muy específicos, los cambios que no afectan a la sintaxis abstracta ni la amplían producen una sintaxis de transferencia compatible hacia atrás cuando se utilizan las reglas de codificación básica. Esos cambios figuran en 12.5.1.1 y 12.5.1.2.

Sin embargo, cuando no existen reglas de compatibilidad hacia adelante, sólo se puede garantizar un adecuado interfuncionamiento si los valores ampliados (es decir, los valores que pertenecen a la sintaxis abstracta ampliada pero no a la original) no se envían nunca a una realización que sólo admite el protocolo original.

Los cambios no compatibles son aquellos que afectan la sintaxis de transferencia de forma no compatible. En este caso, una PDU codificada del protocolo original no es necesariamente una PDU codificada válida para el nuevo protocolo.

En general, un cambio no compatible desde el punto de vista de una sintaxis abstracta produce un cambio no compatible desde el punto de vista de la sintaxis de transferencia. No obstante, puede haber algunas excepciones para un conjunto dado de reglas de codificación.

En 12.5.1.3 figuran algunos ejemplos de esos cambios.

Además, es evidente que, en muchos casos la modificación de las reglas de codificación produce incompatibilidad desde el punto de vista de la sintaxis de transferencia.

12.5.1.1 Cambios sin repercusiones en la sintaxis abstracta

Estos cambios se limitan estrictamente a la forma en que se especifican la sintaxis abstracta y el tipo utilizado para su definición, y no afectan el conjunto de valores definidos por la sintaxis abstracta. Esos cambios pueden ser necesarios para armonizar una especificación con las reglas indicadas en 12.5.1.2 y 12.5.1.3 de este addendum (esa lista no está necesariamente completa).

- a) En un tipo de conjunto o tipo de secuencia, sustituir la utilización de «COMPONENTS OF» con la inclusión directa de los componentes equivalentes, o viceversa.
- b) En un tipo de opción, sustituir los tipos de opción anidados con la inclusión directa de cada «NamedType» (tipo denominado) que aparece en la «AlternativeTypeList» (lista de tipos alternativos).
- c) Sustituir un tipo por una «typereference» (referencia de tipo) que representa el mismo tipo, o viceversa.
- d) Sustituir un valor por un «valuereference» (referencia de valor) que lo representa o viceversa: esto incluye la sustitución de un «number» (número) por una «valuereference».
- e) Sustituir un tipo por un tipo de selección equivalente o viceversa.
- f) Añadir o suprimir (si no se utiliza) uno o más «NamedBit» (bit denominado) de un tipo de cadena de bits.
- g) Añadir o suprimir (si no se utiliza) uno o más «NamedNumber» (número denominado) de un tipo de entero.
- h) Cambiar sistemáticamente la ortografía de «typereference», «modulereference» (referencia de módulo), «valuereference» o «identifier» (identificador) en todos los módulos ASN.1. Esto incluye la adición de identificadores de la sintaxis cuando se permita.
- i) Dividir un módulo ASN.1 en varios módulos ASN.1.
- j) Reunir varios módulos ASN.1 en un módulo ASN.1.
- k) Trasladar partes de un módulo ASN.1 a otro módulo ASN.1.
- l) Añadir uno o más «Symbol» (símbolo) a la lista «EXPORTS». (O suprimir el enunciado «EXPORTS» para indicar que todo se exporta).
- m) Añadir uno o más «Symbol» a la lista «IMPORTS» (símbolos de módulos ASN.1 que ya aparecen referenciados en la lista «IMPORTS» así como símbolos de módulos ASN.1 referenciados nuevamente).
- n) Suprimir uno o más «Valueassignment» (asignación de valor) existentes si su «valuereference» asociada nunca se ha utilizado (ni siquiera implícitamente en la construcción ANY DEFINED BY) en todos los módulos ASN.1.
- o) Suprimir uno o más «Typeassignment» (asignación de tipo) existentes (incluidas aquellas del tipo OPERATION y ERROR) si nunca se han utilizado en todos los módulos ASN.1.
- p) Añadir un tipo o valor ERROR ya incluido en la sintaxis abstracta (es decir, añadido a otro tipo OPERATION) a la lista de ERRORS de un tipo OPERATION si tenía esa lista o añadir una lista de ERRORS a un tipo OPERATION si no tenía una lista.
- q) Añadir un tipo o valor OPERATION ya incluido en la sintaxis abstracta (por ejemplo, no como una Operación enlazada) a la lista de LINKED OPERATIONS de un tipo OPERATION si tenía esa lista o añadir una lista de LINKED OPERATIONS a un tipo OPERATION si no tenía una lista.

12.5.1.2 Ampliación de una sintaxis abstracta

Se amplía una sintaxis abstracta si se amplía su tipo asociado (es decir, si se trata de un tipo de opción, se puede ampliar añadiendo un nuevo componente o ampliando uno existente). Una forma de ampliar una PDU (o cualquier tipo estructurado) es ampliar el tipo de cualquiera de sus componentes.

Se considera que un tipo ASN.1 es una ampliación de otro si el primero incluye todos los valores del segundo, y posiblemente algunos otros.

Dado un cierto tipo, sus ampliaciones son aquellos tipos que se podrían derivar por uno o más de los siguientes cambios, combinados con cualquier número de los descritos en la sección 12.5.1.1.

- a) Cambiar un tipo simple en un tipo de opción que incluya ese tipo en la «AlternativeTypeList».
NOTA 1 – El rótulo de esta alternativa ha de permanecer inalterado, no se permite ningún rótulo (EXPLICIT) adicional. Debe procurarse que todas las referencias a este tipo cambiado en todos los módulos ASN.1 sigan cumpliendo todos los requisitos ASN.1, en particular, que los rótulos sean distintos y los identificadores únicos.
 - b) Añadir uno o más «NamedType» a la «AlternativeTypeList» de un tipo de opción.
NOTA 2 – Véase la nota relativa al cambio de un tipo simple en un tipo de opción.
 - c) Añadir un componente opcional a un tipo de secuencia o a un tipo de conjunto.
 - d) Añadir un componente por defecto a un tipo de secuencia o a un tipo de conjunto.
 - e) Ampliar uno o más componentes de un tipo de opción, tipo de secuencia o tipo de conjunto.
 - f) Ampliar el tipo con respecto al cual se define un tipo de secuencia o un tipo de conjunto.
 - g) Cambiar un componente obligatorio de un tipo de secuencia o tipo de conjunto a un componente opcional o por defecto.
NOTA 3 – El rótulo de este componente permanece inalterado y se debe asegurar la distinción de los rótulos.
 - h) Añadir uno o más nuevos «NamedNumber» a un tipo enumerado.
NOTA 4 – Se debe garantizar la distinción de valores y la unicidad de los identificadores.
 - i) Ampliar el «ValueRange» (gama de valores) de un tipo entero disminuyendo el «LowerEndpoint» (punto extremo inferior) y/o aumentando el «UpperEndpoint» (punto extremo superior).
 - j) Ampliar el «SizeConstraint» (constricción de tamaño) de un tipo de cadena de octetos, un tipo de cadena de bits o un tipo de cadena de caracteres disminuyendo el «LowerEndpoint» y/o aumentando el «UpperEndpoint».
 - k) Ampliar el «SizeConstraint» de un tipo de secuencia o un tipo «conjunto-de» disminuyendo el «LowerEndpoint» y/o aumentando el «UpperEndpoint».
 - l) Cambiar el «Value» (valor) asignado a una «valuereference» si el efecto para todas las referencias cumple aún todas las otras reglas (por ejemplo, aumento de un valor utilizado sólo como «UpperEndpoint» en un «ValueRange» o «SizeConstraint»).
- Los siguientes cambios de la definición OPERATION y ERROR afectan a la sintaxis abstracta formada por el conjunto de valores cuyo tipo es el de los mensajes TCAP o las PDU ROSE parametrizados por una lista específica de operaciones.
- m) Añadir una nueva definición de valor de tipo OPERATION y ERROR, respectivamente, mientras sean distintas de otras definiciones de valor de tipo OPERATION y ERROR, respectivamente.
 - n) Añadir un tipo como ARGUMENT/PARAMETER a un tipo OPERATION si no tenía un ARGUMENT/PARAMETER.
 - o) Añadir un tipo como RESULT a un tipo OPERATION si tenía un RESULT vacío o ningún RESULT.
 - p) Añadir un tipo como PARAMETER a un tipo ERROR si no tenía un PARAMETER.

12.5.1.3 Cambios no compatibles

Los cambios producen incompatibilidad desde el punto de vista de la sintaxis abstracta cuando un valor de la sintaxis abstracta original no es un valor válido para la nueva sintaxis abstracta.

Un cambio no compatible al tipo o tipos con respecto a los cuales se define una sintaxis abstracta produce una incompatibilidad entre la sintaxis abstracta original y la nueva.

En la lista que figura a continuación aparecen algunos ejemplos de esos cambios no compatibles:

- sustituir un tipo por otro aunque el rótulo siga siendo el mismo;
- suprimir una definición de tipo o una definición de valor al que se hace referencia explícita (IMPORTS) o implícitamente (ANY DEFINED BY de TCAP);
- suprimir un «NamedType» desde la «AlternativeTypeList» de un tipo de opción;
- suprimir un «NamedNumber» de «Enumeration» (enumeración) de un tipo enumerado;
- limitar el «ValueRange» de un tipo de entero;
- limitar el «SizeConstraint» de un tipo de cadena;
- limitar el «SizeConstraint» de un tipo «secuencia-de»;
- cambiar el orden de los elementos en la «ElementTypeList» de un tipo de secuencia;
- hacer cualquier combinación de los cambios mencionados de uno o más componentes de un tipo estructurado.

12.5.2 Reglas de compatibilidad hacia adelante

En la mayoría de los casos, la compatibilidad hacia adelante se alcanza mediante negociación del contexto de aplicación. Sin embargo, para minimizar el número de repliegues de protocolo en la red de señalización a veces será necesario definir reglas de compatibilidad hacia adelante que permitan una versión de un protocolo aceptar unidades de datos de protocolo generadas por una futura versión sin tener que suministrar un nuevo nombre de contexto de aplicación.

Esto es necesario también cuando no se admite negociación de contexto de aplicación, por ejemplo, cuando no se admite una parte de diálogo opcional de TC.

Si el diseñador de protocolo desea asegurar de que el valor de una PDU de la nueva versión de un protocolo sea siempre (al menos) parcialmente reconocida por una realización de una versión más antigua de ese protocolo, deben seguirse las siguientes directrices:

- la nueva versión del protocolo debe cumplir las reglas descritas en 12.5.1 (es decir, la nueva sintaxis abstracta será una ampliación de la anterior);
- las reglas de codificación aplicables (permanecerán inalteradas) permiten delimitar las partes desconocidas de la codificación¹⁾;
- deberán incluirse reglas de extensibilidad en la especificación del protocolo original.

Si no se sigue la última recomendación, el comportamiento de una entidad de recepción dependerá de la realización.

Se recomienda limitar el uso de las reglas de extensibilidad a:

- la adición de componentes OPTIONAL y DEFAULT en tipos derivados del tipo SEQUENCE o SET;
- la adición de asignación de bits en tipos derivados del tipo BIT STRING (sin ampliar la restricción de tamaño);
- la adición de alternativas a un tipo CHOICE, siempre que no corresponda a un elemento obligatorio de una estructura superior;
- la adición de valores enumerados a un tipo ENUMERATED, siempre que no corresponda a un elemento obligatorio de una estructura superior.

El diseñador del protocolo debe saber que las reglas de extensibilidad que no figuran en esta lista (por ejemplo, la relajación de una restricción de tamaño o una gama de valor, o la ampliación de un tipo CHOICE correspondiente a un elemento obligatorio) pueden requerir una atención especial (por ejemplo, la especificación de códigos de error que deben utilizarse cuando aparece un elemento de información no reconocido) para evitar importantes errores funcionales.

¹⁾ Esto ocurrirá siempre si se emplean las reglas de codificación básica.

Hay dos formas básicas de suministrar una especificación de las reglas de extensibilidad:

- i) La especificación incluye un enunciado general que se aplica globalmente a cualquier PDU del protocolo.
Esto se ilustra en el siguiente extracto de la Recomendación UIT-T X.862, sobre procesamiento de transacciones:
«Para proporcionar la compatibilidad futura, una TPPM receptora:
 - a) omitirá cualquier elemento no definido de SET, ENUMERATED o SEQUENCE;
 - b) cuando se utilizan bits denominados en ASN.1, tratará cualquier bit como no significativo si no tiene un nombre asignado.»
- ii) La especificación incluye varios enunciados, cada uno de los cuales se aplica específicamente a una PDU particular (o a cualquier estructura interna) del protocolo.

La versión de ASN.1 de 1993 suministra una nueva pieza de notación (bandera) para indicar los tipos que se pueden ampliar de manera que se omitan las adiciones no conocidas. Esta bandera es una elipsis (...) y se denomina un marcador de extensiones. Se recomienda que los usuarios de la versión de ASN.1 de 1988 incluyan esta bandera como un comentario de ASN.1.

Los ejemplos siguientes ilustran una especificación de 1993 en la que se aceptarán valores de elementos suplementarios (no conocidos) para PDU-A y TypeA, pero no para TypeB.

```
PDU-A ::= SEQUENCE {
    element1 TypeA,
    element2 TypeB,
    ...}

TypeA ::= SEQUENCE {
    element3 TypeC,
    element4 TypeD,
    ...
}

TypeB ::= SEQUENCE {
    element5 TypeE,
    element6 TypeF }
```

Puede también ser necesario el mecanismo de extensibilidad cuando hay retransmisiones de la capa de aplicación que admiten una versión del protocolo más baja que la utilizada por los emisores y receptores reales de los mensajes (por ejemplo, diálogo Vn-Vn TC retransmitido por un nodo Vn-1). Sin embargo, en este caso, existe el requisito adicional de que el nodo de retransmisión pase la información no reconocida recibida al nodo siguiente.

La especificación de las reglas de extensibilidad sin ninguna otra indicación no impone ningún requisito a la entidad receptora con respecto al ulterior procesamiento de la parte de la codificación que corresponde a los valores no conocidos. Si no existe ninguna especificación adicional, se supone que se omiten las partes no codificadas.

Si es necesario adoptar alguna otra medida especial (por ejemplo, devolver a un error específico, retransmitir esa parte de la codificación a una tercera parte ...) el diseñador de protocolo debe suministrar una especificación explícita del comportamiento esperado además de las reglas de extensibilidad. Como opción posible, se puede incluir en la versión original del protocolo algunos elementos de información que transmitan en forma dinámica una indicación del comportamiento que debe seguirse con respecto a la recepción de elementos de información no reconocidos.