



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

X.711

(10/97)

SERIE X: REDES DE DATOS Y COMUNICACIÓN
ENTRE SISTEMAS ABIERTOS

Gestión de interconexión de sistemas abiertos – Servicio y
protocolo de comunicación de gestión

**Tecnología de la información – Interconexión de
sistemas abiertos – Protocolo común de
información de gestión: Especificación**

Recomendación UIT-T X.711

(Anteriormente Recomendación del CCITT)

RECOMENDACIONES DE LA SERIE X DEL UIT-T
REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS

REDES PÚBLICAS DE DATOS	
Servicios y facilidades	X.1–X.19
Interfaces	X.20–X.49
Transmisión, señalización y conmutación	X.50–X.89
Aspectos de redes	X.90–X.149
Mantenimiento	X.150–X.179
Disposiciones administrativas	X.180–X.199
INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Modelo y notación	X.200–X.209
Definiciones de los servicios	X.210–X.219
Especificaciones de los protocolos en modo conexión	X.220–X.229
Especificaciones de los protocolos en modo sin conexión	X.230–X.239
Formularios para declaraciones de conformidad de implementación de protocolo	X.240–X.259
Identificación de protocolos	X.260–X.269
Protocolos de seguridad	X.270–X.279
Objetos gestionados de capa	X.280–X.289
Pruebas de conformidad	X.290–X.299
INTERFUNCIONAMIENTO ENTRE REDES	
Generalidades	X.300–X.349
Sistemas de transmisión de datos por satélite	X.350–X.399
SISTEMAS DE TRATAMIENTO DE MENSAJES	X.400–X.499
DIRECTORIO	X.500–X.599
GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS	
Gestión de redes	X.600–X.629
Eficacia	X.630–X.639
Calidad de servicio	X.640–X.649
Denominación, direccionamiento y registro	X.650–X.679
Notación de sintaxis abstracta uno	X.680–X.699
GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Marco y arquitectura de la gestión de sistemas	X.700–X.709
Servicio y protocolo de comunicación de gestión	X.710–X.719
Estructura de la información de gestión	X.720–X.729
Funciones de gestión y funciones de arquitectura de gestión distribuida abierta	X.730–X.799
SEGURIDAD	X.800–X.849
APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Cometimiento, concurrencia y recuperación	X.850–X.859
Procesamiento de transacciones	X.860–X.879
Operaciones a distancia	X.880–X.899
PROCESAMIENTO DISTRIBUIDO ABIERTO	X.900–X.999

NORMA INTERNACIONAL 9596-1

RECOMENDACIÓN UIT-T X.711

**TECNOLOGÍA DE LA INFORMACIÓN – INTERCONEXIÓN DE SISTEMAS
ABIERTOS – PROTOCOLO COMÚN DE INFORMACIÓN
DE GESTIÓN: ESPECIFICACIÓN**

Orígenes

El texto de la Recomendación UIT-T X.711 se aprobó el 24 de octubre de 1997. Su texto se publica también, en forma idéntica, como Norma Internacional ISO/CEI 9596-1.

PREFACIO

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

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

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

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

NOTA

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

PROPIEDAD INTELECTUAL

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

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

© UIT 1998

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

ÍNDICE

	<i>Página</i>
1	Campo de aplicación 1
2	Referencias normativas 1
2.1	Recomendaciones Normas Internacionales idénticas 1
2.2	Pares de Recomendaciones Normas Internacionales de contenido técnico equivalente..... 2
3	Definiciones 2
3.1	Definiciones del modelo de referencia básico..... 2
3.2	Definiciones del marco de gestión 2
3.3	Definiciones de operaciones a distancia 3
3.4	Definiciones del servicio común de información de gestión..... 3
3.5	Definiciones de elemento de servicio de control de asociación (ACSE) 3
3.6	Definiciones de presentación 3
4	Símbolos y abreviaturas 3
5	Visión de conjunto 4
5.1	Servicio prestado 4
5.2	Servicios subyacentes 4
5.3	Definiciones de información de gestión..... 5
5.4	Versión de protocolo..... 5
6	Elementos de procedimiento 5
6.1	Establecimiento de la asociación 5
6.2	Operaciones a distancia 5
6.3	Procedimiento de informe de evento..... 6
6.4	Procedimiento de obtención..... 7
6.5	Procedimiento de fijación 8
6.6	Procedimiento de acción 9
6.7	Procedimiento de creación 10
6.8	Procedimiento de supresión 10
6.9	Liberación ordenada de asociación 11
6.10	Liberación abrupta de asociación..... 11
7	Sintaxis abstracta..... 11
7.1	Convenios 11
7.2	Correspondencia entre primitivas de CMISE y operaciones de CMIP 12
7.3	Datos de usuario de ACSE..... 13
7.4	Unidades de datos CMIP 14
7.5	Definición de sintaxis abstracta para CMIP..... 21
8	Conformidad 22
8.1	Requisitos estáticos..... 22
8.2	Requisitos dinámicos 23
Anexo A	– Reglas de asociación para el CMISE 24
A.1	ACSE, requisitos de sesión y presentación 24
A.2	Reglas de iniciación de asociación..... 24
A.3	Reglas de liberación de asociación 25
A.4	Reglas de aborto de asociación 25
Anexo B	– Sintaxis ASN.1 ampliada 27
Anexo C	– Ejemplos de APDU de ROSE del CMISE 35
Anexo D	– Sintaxis abstracta de guía 36

NORMA INTERNACIONAL

RECOMENDACIÓN UIT-T

TECNOLOGÍA DE LA INFORMACIÓN – INTERCONEXIÓN DE SISTEMAS ABIERTOS – PROTOCOLO COMÚN DE INFORMACIÓN DE GESTIÓN: ESPECIFICACIÓN

1 Campo de aplicación

Esta Recomendación | Norma Internacional especifica un protocolo utilizado por las entidades de capa de aplicación para intercambiar información de gestión.

Esta Recomendación | Norma Internacional especifica:

- los procedimientos para la transmisión de información de gestión entre entidades de aplicación;
- la sintaxis abstracta del protocolo común de información de gestión (CMIP, *common management information protocol*) y las reglas de codificación asociadas que han de aplicarse;
- los procedimientos para la interpretación correcta de información de control de protocolo;
- los requisitos de conformidad que ha de satisfacer la implementación de esta Recomendación.

Esta Recomendación | Norma Internacional no especifica:

- la estructura o significado de la información de gestión transmitida por medio del CMIP;
- la manera en que se realiza la gestión como resultado de intercambios de CMIP;
- las interacciones que resultan de la utilización del CMIP.

2 Referencias normativas

Las siguientes Recomendaciones y Normas Internacionales contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación | Norma Internacional. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y Normas son objeto de revisiones, por lo que se preconiza que los participantes en acuerdos basados en la presente Recomendación | Norma Internacional investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y las Normas citadas a continuación. Los miembros de la CEI y de la ISO mantienen registros de las Normas Internacionales actualmente vigentes. La Oficina de Normalización de las Telecomunicaciones de la UIT mantiene una lista de las Recomendaciones UIT-T actualmente vigentes.

2.1 Recomendaciones | Normas Internacionales idénticas

- Recomendación UIT-T X.200 (1994) | ISO/CEI 7498-1:1994, *Tecnología de la información – Interconexión de sistemas abiertos – Modelo de referencia básico: El modelo básico*.
- Recomendación UIT-T X.215 (1995) | ISO/CEI 8326:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Definición del servicio de sesión*.
- Recomendación UIT-T X.216 (1994) | ISO/CEI 8822:1994, *Tecnología de la información – Interconexión de sistemas abiertos – Definición del servicio de presentación*.
- Recomendación UIT-T X.217 (1995) | ISO/CEI 8649:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Definición de servicio para el elemento de servicio de control de asociación*.
- Recomendación UIT-T X.226 (1994) | ISO/CEI 8823-1:1994, *Tecnología de la información – Interconexión de sistemas abiertos – Protocolo de presentación con conexión: Especificación del protocolo*.

- Recomendación UIT-T X.227 (1995) | ISO/CEI 8650-1:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Protocolo con conexión para el elemento de servicio de control de asociación: Especificación de protocolo.*
- Recomendación UIT-T X.710 (1997) | ISO/CEI 9595:1998, *Tecnología de la información – Interconexión de sistemas abiertos – Servicio común de información de gestión.*
- Recomendación X.712 del CCITT (1992) | ISO/CEI 9596-2:1993, *Tecnología de la información – Interconexión de sistemas abiertos – Protocolo común de información de gestión: Formulario de enunciado de conformidad de implementación de protocolo.*

2.2 Pares de Recomendaciones | Normas Internacionales de contenido técnico equivalente

- Recomendación X.208 del CCITT (1988), *Especificación de la notación de sintaxis abstracta uno.*
ISO/CEI 8824:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1).*
- Recomendación X.209 del CCITT (1988), *Especificación de las reglas básicas de codificación de la notación de sintaxis abstracta uno.*
ISO/CEI 8825:1990, *Information technology – Open systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).*
- Recomendación X.219 del CCITT (1988), *Operaciones a distancia: Modelo, notación y definición del servicio.*
ISO/CEI 9072-1:1989, *Information processing systems – Text communication – Remote Operations – Part 1: Model, notation and service definition.*
- Recomendación X.229 del CCITT (1988), *Operaciones a distancia: Especificación del protocolo.*
ISO/CEI 9072-2:1989, *Information processing systems – Text communication – Remote Operations – Part 2: Protocol specification.*
- Recomendación X.700 del CCITT (1992), *Marco de gestión para la interconexión de sistemas abiertos para aplicaciones del CCITT.*
ISO/CEI 7498-4:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 4: Management framework.*

3 Definiciones

A los efectos de esta Recomendación | Norma Internacional se aplican las definiciones siguientes.

3.1 Definiciones del modelo de referencia básico

En esta Recomendación | Norma Internacional se utilizan los siguientes términos definidos en la Rec. UIT-T X.200 | ISO/CEI 7498-1:

- a) elemento de servicio de aplicación;
- b) proceso de aplicación;
- c) sistema abierto real;
- d) gestión de sistemas.

3.2 Definiciones del marco de gestión

En esta Recomendación | Norma Internacional se utilizan los siguientes términos definidos en la Rec. X.700 del CCITT | ISO/CEI 7498-4:

- a) objeto gestionado;
- b) información de gestión;
- c) base de información de gestión;
- d) entidad de aplicación de gestión de sistemas.

3.3 Definiciones de operaciones a distancia

En esta Recomendación | Norma Internacional se utilizan los siguientes términos definidos en la Rec. X.219 del CCITT | ISO/CEI 9072-1:

- a) iniciador de asociación;
- b) respondedor de asociación;
- c) operaciones ligadas;
- d) operaciones a distancia;
- e) elemento de servicio de operaciones a distancia;
- f) invocador;
- g) realizador;
- h) clase de asociación;
- i) clase de operación.

3.4 Definiciones del servicio común de información de gestión

En esta Recomendación | Norma Internacional se utilizan los siguientes términos definidos en la Rec. UIT-T X.710 | ISO/CEI 9595:

- a) atributo;
- b) elemento de servicio común de información de gestión;
- c) servicio común de información de gestión;
- d) proveedor del servicio CMISE;
- e) usuario del servicio CMISE;
- f) usuario invocador del servicio CMISE;
- g) usuario realizador del servicio CMISE.

3.5 Definiciones de elemento de servicio de control de asociación (ACSE)

En esta Recomendación | Norma Internacional se utilizan los siguientes términos definidos en la Rec. UIT-T X.217 | ISO/CEI 8649:

- a) contexto de aplicación;
- b) asociación de aplicación;
- c) asociación.

3.6 Definiciones de presentación

En esta Recomendación | Norma Internacional se utilizan los siguientes términos definidos en la Rec. UIT-T X.216 | ISO/CEI 8822:

- a) sintaxis abstracta;
- b) sintaxis de transferencia.

4 Símbolos y abreviaturas

A los efectos de esta Recomendación | Norma Internacional se utilizan las siguientes siglas.

ACSE	Elemento de servicio de control de asociación (<i>association control service element</i>)
APDU	Unidad de datos de protocolo de aplicación (<i>application protocol data unit</i>)
ASE	Elemento de servicio de aplicación (<i>application service element</i>)
ASN.1	Notación de sintaxis abstracta uno (<i>abstract syntax notation one</i>)
CMIP	Protocolo común de información de gestión (<i>common management information protocol</i>)
CMIPM	Máquina de protocolo común de información de gestión (<i>common management information protocol machine</i>)

CMIS	Servicio común de información de gestión (<i>common management information services</i>)
CMISE	Elemento de servicio común de información de gestión (<i>common management information service element</i>)
DCS	Conjunto de contexto definido (<i>defined context set</i>)
PCI	Información de control de protocolo (<i>protocol control information</i>)
PDU	Unidad de datos de protocolo (<i>protocol data unit</i>)
PICS	Enunciado de conformidad de implementación de protocolo (<i>protocol implementation conformance statement</i>)
RO	Operaciones a distancia (<i>remote operations</i>)
ROSE	Elemento de servicio de operaciones a distancia (<i>remote operations service element</i>)
SMAE	Entidad de aplicación de gestión de sistemas (<i>systems management application-entity</i>)

5 Visión de conjunto

El protocolo común de información de gestión (CMIP) especifica elementos de protocolo que pueden utilizarse para proporcionar los servicios de operaciones y notificación descritos en la Rec. UIT-T X.710 | ISO/CEI 9595, que define los servicios comunes de información de gestión (CMIS).

5.1 Servicio prestado

El protocolo especificado en esta Recomendación | Norma Internacional soporta los servicios especificados en la Rec. UIT-T X.710 | ISO/CEI 9595. Estos servicios se resumen en el cuadro 1.

Cuadro 1 – Servicios comunes de información de gestión

Servicio	Tipo
M-CANCELACIÓN-OBTENCIÓN	Confirmado
M-INFORME-EVENTO	Confirmado/no confirmado
M-OBTENCIÓN	Confirmado
M-FIJACIÓN	Confirmado/no confirmado
M-ACCIÓN	Confirmado/no confirmado
M-CREACIÓN	Confirmado
M-SUPRESIÓN	Confirmado

5.2 Servicios subyacentes

En esta Recomendación | Norma Internacional se utilizan los servicios RO-INVOCACIÓN, RO-RESULTADO, RO-ERROR y RO-RECHAZO-U del elemento de servicio de operaciones a distancia (ROSE, *remote operations service element*) definido en la Rec. X.219 del CCITT | ISO/CEI 9072-1. El ROSE conlleva la utilización del servicio de presentación definido en la Rec. UIT-T X.216 | ISO/CEI 8822. Las operaciones confirmadas del CMIP son operaciones de clase 2 (asíncronas) u operaciones de clase 1 (síncronas), según lo requiera la aplicación. La elección de la clase de operaciones es un asunto local. Las operaciones no confirmadas del CMIP son las operaciones de clase 5 (asíncronas, resultado no comunicado). El CMIP utiliza asociaciones de clase 3.

Si la unidad funcional servicio ampliado se negocia satisfactoriamente, las unidades de datos de protocolo de aplicación del ROSE pueden hacerse corresponder con otros servicios de presentación distintos del servicio P-DATOS.

NOTA – Por ejemplo, puede ser necesario modificar el conjunto de contexto definido (DCS, *defined context set*) definido por la presentación cuando la operación CMIP se envía al usuario del servicio CMISE. En este caso, la APDU de ROSE que transporta la operación CMIP se hará corresponder con el servicio P-MODIFICACIÓN-CONTEXTO que también se utiliza para realizar los cambios del DCS.

En la descripción del contexto de aplicación utilizado en la asociación se dan detalles de los otros servicios de presentación que se necesitan y de la manera de utilizarlos.

5.2.1 Servicios que se supone prestará el ACSE

En esta Recomendación | Norma Internacional se supone la utilización de los servicios A-ASOCIACIÓN, A-LIBERACIÓN, A-ABORTO y A-P-ABORTO del elemento de servicio de control de asociación.

5.2.2 Servicios de la capa de presentación cuya utilización se supone

En la Rec. X.229 del CCITT | ISO/CEI 9072-2 se supone la utilización del servicio P-DATOS de la capa de presentación para la transferencia de las PDU RO-INVOCACIÓN, RO-RESULTADO, RO-ERROR y RO-RECHAZO.

5.3 Definiciones de información de gestión

En esta Recomendación | Norma Internacional se define la sintaxis abstracta del protocolo común de información de gestión. Las definiciones de información de gestión transportadas por el protocolo no se especifican en la presente Recomendación | Norma Internacional.

5.4 Versión de protocolo

En la presente Recomendación | Norma Internacional se define la versión 2 del CMIP. La versión 2 sustituye a la versión 1. Esta Recomendación | Norma Internacional no define ningún interfuncionamiento entre la versión 2 y la versión 1.

6 Elementos de procedimiento

En esta cláusula se definen los elementos de procedimiento del CMIP. Los procedimientos definen la transferencia de las PDU de CMIP cuya estructura, codificación y relación con las primitivas de servicio del CMIS se especifican en la cláusula 7.

La máquina de protocolo común de información de gestión (CMIPM, *common management information protocol machine*) acepta primitivas de servicio petición y respuesta del CMIS, y emite PDU de CMIP que inician elementos de procedimiento específicos descritos en este punto.

Una CMIPM aceptará cualquier PDU de CMIP bien formada y la pasará al usuario realizador del servicio CMISE para su procesamiento, por medio de las primitivas de servicio indicación y confirmación del CMIS. Si la PDU recibida no está bien formada, o no contiene una notificación u operación soportada, se devuelve una PDU que indica que la PDU recibida ha sido rechazada.

Los procedimientos indican solamente cómo deben interpretarse los diversos campos de la PDU del CMIP, y no lo que un usuario invocador del servicio CMIS debe hacer con la información que pide, ni cómo un usuario realizador del servicio CMIS debe procesar la invocación.

6.1 Establecimiento de la asociación

El establecimiento de una asociación comprende dos usuarios del servicio CMIS, uno que es el iniciador de la asociación y el otro que es el respondedor de la asociación.

Un usuario del servicio CMIS puede iniciar un establecimiento de asociación utilizando el servicio A-ASOCIACIÓN de la Rec. UIT-T X.217 | ISO/CEI 8649.

El contexto de aplicación específica, entre otras cosas, las reglas requeridas para la coordinación de la información de iniciación correspondiente a distintos ASE. Las reglas de asociación del CMIS se especifican en el anexo A.

6.2 Operaciones a distancia

6.2.1 Elementos de procedimiento de operaciones a distancia

Los elementos de procedimiento del CMIP dependen de los siguientes elementos de procedimiento de operaciones a distancia subyacentes:

- a) invocación;
- b) retorno de resultado;

- c) retorno de error;
- d) rechazo de usuario;
- e) rechazo de proveedor.

Estos elementos de procedimiento se describen detalladamente en la Rec. X.229 del CCITT | ISO/CEI 9072-2.

En el cuadro 2 se especifica la correspondencia entre los parámetros CMIS y ROSE.

Cuadro 2 – Correspondencia entre los parámetros CMIS y ROSE

Parámetro CMIS	Parámetro ROSE
Identificador de invocación	ID de invocación
Identificador de ligazón	ID de ligazón

La correspondencia entre otros parámetros CMIS y ROSE se especifica en la cláusula 7.

6.2.2 Parámetros de problema RO-Rechazo

Los parámetros de problema RO-Rechazo se hacen corresponder o se procesan como sigue.

6.2.2.1 La correspondencia de RO-rechazo por usuario. Problema de invocación (RO-Reject-User.Invoke-problem) con códigos de error CMIS se indican en el cuadro 3.

Cuadro 3 – Correspondencia de RO-Rechazo-usuario. Invocación-problema con códigos de error CMISE

Parámetro RO-RECHAZO	Código de error CMISE
duplicate-invocation	invocación duplicada
mistyped-argument	argumento mal tipificado
resource-limitation	limitación de recurso
unrecognized-operation	operación no reconocida

Los demás parámetros de problema de invocación son un asunto local.

6.2.2.2 Los demás parámetros RO-Rechazo se tratarán como un asunto local.

6.3 Procedimiento de informe de evento

6.3.1 Invocación

Los procedimientos de informe de evento son iniciados por la primitiva petición M-INFORME-EVENTO.

Al recibir la primitiva petición M-INFORME-EVENTO, la CMIPM:

- a) construirá en el modo confirmado, una APDU que solicite la operación, m-Informe-Evento-Confirmado (m-EventReport-Confirmed), y en caso contrario, construirá una APDU que solicite la operación m-Informe-Evento (m-EventReport);
- b) enviará la APDU empleando el procedimiento RO-INVOCACIÓN.

6.3.2 Recepción

Al recibir una APDU que solicita la operación m-EventReport o m-EventReport-Confirmed, la CMIPM emitirá, si la APDU está bien formada, una primitiva indicación M-INFORME-EVENTO al usuario del servicio CMISE, con el parámetro modo indicando si se solicita o no confirmación, y en caso contrario, construirá una APDU que contenga una notificación del error y la enviará empleando el procedimiento RO-RECHAZO-U.

6.3.3 Respuesta

En el modo confirmado, la CMIPM aceptará una primitiva respuesta M-INFORME-EVENTO, y:

- a) construirá una APDU que confirme la notificación M-INFORME-EVENTO;
- b) si los parámetros de la primitiva respuesta M-INFORME-EVENTO indican que la notificación ha sido aceptada, enviará la APDU empleando el procedimiento RO-RESULTADO, y en caso contrario, enviará la APDU empleando el procedimiento RO-ERROR.

6.3.4 Recepción de respuesta

Al recibir una APDU que responde a una notificación M-INFORME-EVENTO, la CMIPM emitirá, si la APDU está bien formada, una primitiva confirmación M-INFORME-EVENTO al usuario del servicio CMISE, completando así el procedimiento de notificación, y en caso contrario, construirá una APDU que contenga una notificación del error y la enviará empleando el procedimiento RO-RECHAZO-U.

6.4 Procedimiento de obtención

6.4.1 Invocación

Los procedimientos de obtención son iniciados por la primitiva petición M-OBTENCIÓN.

Al recibir la primitiva petición M-OBTENCIÓN, la CMIPM:

- a) construirá una APDU que solicite la operación m-Obtención;
- b) enviará la APDU empleando el procedimiento RO-INVOCACIÓN.

6.4.2 Recepción

Al recibir una APDU que solicita la operación m-Obtención, la CMIPM enviará, si la APDU está bien formada, una primitiva indicación M-OBTENCIÓN al usuario del servicio CMISE, y en caso contrario, construirá una APDU que contiene una notificación del error y la enviará empleando el procedimiento RO-RECHAZO-U.

6.4.3 Respuesta

La CMIPM:

- a) aceptará cero o más primitivas respuesta M-OBTENCIÓN que contengan un ID de ligazón seguido por una sola primitiva respuesta M-OBTENCIÓN sin ID de ligazón;
- b) para cada primitiva respuesta M-OBTENCIÓN que contiene un ID de ligazón:
 - construirá una APDU que solicite la operación m-Respuesta-Ligada (m-Linked-Reply) con LinkedReplyArgument fijado de manera apropiada a getListError, a getResult o a processingFailure;
 - enviará cada APDU empleando el procedimiento RO-INVOCACIÓN;
- c) para la primitiva respuesta M-OBTENCIÓN que no contenga un ID de ligazón, la CMIPM:
 - construirá una APDU que confirme la operación m-Obtención;
 - si los parámetros de la primitiva respuesta M-OBTENCIÓN indican que la operación ha sido realizada correctamente, enviará la APDU empleando el procedimiento RO-RESULTADO. Si los parámetros de la primitiva respuesta M-OBTENCIÓN indican que la operación ha obtenido un éxito parcial, o no se ha realizado a causa de un error, la CMIPM enviará la APDU empleando el procedimiento RO-ERROR.

6.4.4 Recepción de respuesta

Al recibir una APDU que responda a una operación m-Obtención, la CMIPM:

- a) si la APDU incluye un ID de ligazón y, está bien formada, enviará una primitiva confirmación M-OBTENCIÓN al usuario del servicio CMISE;
- b) si la APDU es la última respuesta (es decir que no contiene un ID de ligazón) y está bien formada, enviará una primitiva confirmación M-OBTENCIÓN al usuario del servicio CMISE, completando así el procedimiento M-OBTENCIÓN;
- c) si la APDU no está bien formada, construirá una APDU que contiene una notificación del error y la enviará empleando el procedimiento RO-RECHAZO-U.

6.4.5 Procedimiento de cancelación de obtención

6.4.5.1 Invocación

Los procedimientos de cancelación de obtención (CancelGet) son iniciados por la primitiva petición M-CANCELACIÓN-OBTENCIÓN.

Al recibir la primitiva petición M-CANCELACIÓN-OBTENCIÓN, la CMIPM:

- a) construirá una APDU que solicite la operación m-Cancelación de Obtención;
- b) enviará la APDU empleando el procedimiento RO-INVOCACIÓN.

6.4.5.2 Recepción

Al recibir una APDU que solicita la operación m-Cancelación de Obtención, la CMIPM enviará, si la APDU está bien formada, una primitiva indicación M-CANCELACIÓN-OBTENCIÓN al usuario del servicio CMISE, y en caso contrario, construirá una APDU que contenga una notificación del error y la enviará empleando el procedimiento RO-RECHAZO-U.

6.4.5.3 Respuesta

La CMIPM:

- a) construirá una APDU que confirme la operación m-Cancelación de Obtención;
- b) si los parámetros de la primitiva respuesta M-CANCELACIÓN-OBTENCIÓN indican que la operación se ejecutó correctamente, enviará la APDU utilizando el procedimiento RO-RESULTADO; en otro caso, enviará la APDU utilizando el procedimiento RO-ERROR. Si la operación m-Cancelación de Obtención tiene éxito, el usuario realizador del servicio CMISE dejará de enviar respuestas ligadas a la operación m-Obtención y enviará una primitiva respuesta M-OBTENCIÓN que contendrá el error "operación cancelada".

6.4.5.4 Recepción de respuesta

Al recibir una APDU que responde a una operación m-Cancelación de Obtención, la CMIPM enviará, si la APDU está bien formada, una primitiva confirmación M-CANCELACIÓN-OBTENCIÓN al usuario del servicio CMISE, y en caso contrario, construirá una APDU que contenga una notificación del error y la enviará empleando el procedimiento RO-RECHAZO-U.

6.5 Procedimiento de fijación

6.5.1 Invocación

Los procedimientos de fijación son iniciados por la primitiva petición M-FIJACIÓN.

Al recibir la primitiva petición M-FIJACIÓN, la CMIPM:

- a) en el modo confirmado, construirá una APDU que solicite la operación m-Fijación-Confirmado (m-Set-Confirmed), y en caso contrario, construirá una APDU que solicita la operación m-Fijación;
- b) enviará la APDU empleando el procedimiento RO-INVOCACIÓN.

6.5.2 Recepción

Al recibir una APDU que solicita la operación m-Fijación o m-Fijación-Confirmado, la CMIPM enviará, si la APDU está bien formada, una primitiva indicación M-FIJACIÓN al usuario del servicio CMISE, con el parámetro modo que indica si se solicita o no confirmación, y en caso contrario, construirá una APDU que contiene una notificación del error y la enviará empleando el procedimiento RO-RECHAZO-U.

6.5.3 Respuesta

En el modo confirmado, la CMIPM:

- a) aceptará cero o más primitivas respuesta M-FIJACIÓN que contengan un ID de ligazón seguido por una sola primitiva respuesta M-FIJACIÓN sin ID de ligazón;
- b) para cada primitiva respuesta M-FIJACIÓN que contiene un ID de ligazón:
 - construirá una APDU que solicita la operación m-Respuesta-Ligada (m-Linked-Reply) con LinkedReplyArgument fijado de manera apropiada a setListError, setResult o processingFailure;
 - enviará las APDU empleando el procedimiento RO-INVOCACIÓN;

- c) para la primitiva respuesta M-FIJACIÓN que no contiene ID de ligazón:
 - construirá una APDU que confirma la operación m-Fijación;
 - si los parámetros de la primitiva respuesta M-FIJACIÓN indican que la operación ha sido realizada correctamente, enviará la APDU empleando el procedimiento RO-RESULTADO. Si los parámetros de la primitiva respuesta M-FIJACIÓN indican que la operación ha obtenido un éxito parcial, o no se ha realizado a causa de un error, enviará la APDU empleando el procedimiento RO-ERROR.

6.5.4 Recepción de respuesta

Al recibir una APDU que responde a una operación m-Fijación-Confirmado, la CMIPM:

- a) si la APDU incluye un ID de ligazón y está bien formada, enviará una primitiva confirmación M-FIJACIÓN al usuario del servicio CMISE;
- b) si la APDU es la última respuesta (es decir, no contiene un ID de ligazón) y está bien formada, enviará una primitiva confirmación M-FIJACIÓN al usuario del servicio CMISE, completando así el procedimiento M-FIJACIÓN;
- c) si la APDU no está bien formada, construirá una APDU que contiene una notificación del error y la enviará empleando el procedimiento RO-RECHAZO-U.

6.6 Procedimiento de acción

6.6.1 Invocación

Los procedimientos de acción son iniciados por la primitiva petición M-ACCIÓN.

Al recibir la primitiva petición M-ACCIÓN, la CMIPM:

- a) en el modo confirmado, construirá una APDU que solicita la operación m-Acción-Confirmada (m-Action-Confirmed), y en caso contrario, construirá una APDU que solicita la operación m-Acción (m-Action);
- b) enviará la APDU empleando el procedimiento RO-INVOCACIÓN.

6.6.2 Recepción

Al recibir una APDU que solicita la operación m-Acción o m-Acción-Confirmada (m-Action-Confirmed), la CMIPM enviará, si la APDU está bien formada, una primitiva indicación M-ACCIÓN al usuario del servicio CMISE, con el parámetro modo que indica si se solicita o no confirmación, y en caso contrario, construirá una APDU que contiene la notificación del error y la enviará empleando el procedimiento RO-RECHAZO-U.

6.6.3 Respuesta

En el modo confirmado, la CMIPM:

- a) aceptará cero o más primitivas respuesta M-ACCIÓN que contengan un ID de ligazón seguido por una sola primitiva respuesta M-ACCIÓN sin ID de ligazón;
- b) para cada primitiva respuesta M-ACCIÓN que contiene un ID de ligazón:
 - construirá una APDU que solicite la operación m-Respuesta-Ligada con LinkedReplyArgument fijado de manera apropiada a actionError, actionResult o processingFailure;
 - enviará las APDU empleando el procedimiento RO-INVOCACIÓN;
- c) para la primitiva respuesta M-ACCIÓN que no contiene ID de ligazón:
 - construirá una APDU que confirme la operación m-Acción;
 - si los parámetros de la primitiva respuesta M-ACCIÓN indican que la operación ha sido realizada correctamente, enviará la APDU empleando el procedimiento RO-RESULTADO, y en caso contrario, enviará la APDU empleando el procedimiento RO-ERROR.

6.6.4 Recepción de respuesta

Al recibir una APDU que responde a una operación m-Acción-Confirmada, la CMIPM:

- a) si la APDU incluye un ID enlazado y está bien formada, enviará una primitiva confirmación M-ACCIÓN al usuario del servicio CMISE;
- b) si la APDU es la última respuesta (es decir, no contiene un ID de ligazón) y está bien formada, enviará una primitiva confirmación M-ACCIÓN al usuario del servicio CMISE, completando así el procedimiento M-ACCIÓN;
- c) si la APDU no está bien formada, construirá una APDU que contiene una notificación del error y la enviará empleando el procedimiento RO-RECHAZO-U.

6.7 Procedimiento de creación

6.7.1 Invocación

Los procedimientos de creación son iniciados por la primitiva petición M-CREACIÓN.

Al recibir la primitiva petición M-CREACIÓN, la CMIPM:

- a) construirá una APDU que solicita la operación m-create;
- b) enviará la APDU empleando el procedimiento RO-INVOCACIÓN.

6.7.2 Recepción

Al recibir una APDU que solicita la operación m-Creación (m-Create), la CMIPM, si la APDU está bien formada, enviará una primitiva indicación M-CREACIÓN al usuario del servicio CMISE, y en caso contrario, construirá una APDU que contenga una notificación del error y la enviará empleando el procedimiento RO-RECHAZO-U.

6.7.3 Respuesta

La CMIPM aceptará una primitiva respuesta M-CREACIÓN y:

- a) construirá una APDU que confirma la operación m-Creación;
- b) si los parámetros de la primitiva respuesta M-CREACIÓN indican que la operación ha sido realizada correctamente, enviará la APDU empleando el procedimiento RO-RESULTADO, y en caso contrario, enviará la APDU empleando el procedimiento RO-ERROR.

6.7.4 Recepción de respuesta

Al recibir una APDU que responde a una operación m-Creación, la CMIPM enviará, si la APDU está bien formada, una primitiva confirmación M-CREACIÓN al usuario del servicio CMISE, completando así el procedimiento M-CREACIÓN, y en caso contrario, construirá una APDU que contiene una notificación del error y la enviará empleando el procedimiento RO-RECHAZO-U.

6.8 Procedimiento de supresión

6.8.1 Invocación

Los procedimientos de supresión son iniciados por la primitiva petición M-SUPRESIÓN.

Al recibir la primitiva petición M-SUPRESIÓN, la CMIPM:

- a) construirá una APDU que solicita la operación m-Supresión (m-Delete);
- b) enviará la APDU empleando el procedimiento RO-INVOCACIÓN.

6.8.2 Recepción

Al recibir una APDU que solicita la operación m-Supresión, la CMIPM enviará, si la APDU está bien formada, una primitiva indicación M-SUPRESIÓN al usuario del servicio CMISE, y en caso contrario, construirá una APDU que contenga una notificación del error y la enviará empleando el procedimiento RO-RECHAZO-U.

6.8.3 Respuesta

La CMIPM:

- a) aceptará cero o más primitivas respuesta M-SUPRESIÓN que contengan un ID de ligazón seguido por una sola primitiva respuesta M-SUPRESIÓN sin ID de ligazón;
- b) para cada primitiva respuesta M-SUPRESIÓN que contiene un ID de ligazón;
 - construirá una APDU que solicita la operación m-Respuesta-Ligada con LinkedReplyArgument fijado de manera apropiada a deleteError, deleteResult o processingFailure;
 - enviará las APDU empleando el procedimiento RO-INVOCACIÓN.
- c) para la primitiva respuesta M-SUPRESIÓN que no contiene ID de ligazón:
 - construirá una APDU que confirme la operación m-Supresión;
 - si los parámetros de la primitiva respuesta M-SUPRESIÓN indican que la operación ha sido realizada correctamente, enviará la APDU empleando el procedimiento RO-RESULTADO, y en caso contrario, enviará la APDU empleando el procedimiento RO-ERROR.

6.8.4 Recepción de respuesta

Al recibir una APDU que responde a una operación m-Supresión, la CMIPM:

- a) si la APDU incluye un ID enlazado, emitirá una primitiva confirmación M-SUPRESIÓN al usuario del servicio CMISE;
- b) si la APDU es la última respuesta (es decir, no contiene un ID de ligazón) y está bien formada, enviará una primitiva confirmación M-SUPRESIÓN al usuario del servicio CMISE completando así el procedimiento M-SUPRESIÓN;
- c) si la APDU no está bien formada, construirá una APDU que contiene una notificación del error y la enviará empleando el procedimiento RO-RECHAZO-U.

6.9 Liberación ordenada de asociación

Cualquiera de los dos usuarios del servicio CMISE puede iniciar una liberación ordenada de la asociación utilizando el servicio A-LIBERACIÓN de la Rec. UIT-T X.217 | ISO/CEI 8649.

NOTA – Esta especificación es diferente de la relativa a la utilización por el ROSE de la operación VINCULACIÓN, en la cual únicamente el iniciador de la asociación puede utilizar el procedimiento A-LIBERACIÓN.

6.10 Liberación abrupta de asociación

Cualquiera de los dos usuarios del servicio CMISE puede iniciar una liberación abrupta de la asociación utilizando el servicio A-ABORTO de la Rec. UIT-T X.217 | ISO/CEI 8649.

El usuario del servicio CMISE puede recibir una indicación de liberación abrupta de la asociación vía los servicios A-ABORTO y A-P-ABORTO de la Rec. UIT-T X.217 | ISO/CEI 8649.

7 Sintaxis abstracta¹⁾

En esta cláusula se especifica la sintaxis abstracta para las PDU de CMIP.

7.1 Convenios

La sintaxis abstracta se define utilizando la notación especificada en la Rec. X.208 del CCITT | ISO/CEI 8824. Las producciones MACRO de ASN.1 utilizadas en esta Recomendación | Norma Internacional, o citadas en la misma, no emplean los aspectos ambiguos de la gramática.

¹⁾ Los usuarios de la presente Recomendación | Norma Internacional pueden reproducir libremente las definiciones de sintaxis abstracta de esta cláusula para que puedan aplicarse al fin que deseen.

Para cada uno de los parámetros de servicio CMISE que han de ser transferidos por una PDU de CMIP, hay un campo de PDU (un ASN.1 NamedType) con el mismo nombre que el parámetro de servicio correspondiente (véase la Rec. UIT-T X.710 | ISO/CEI 9595), salvo las diferencias requeridas por la utilización de la ASN.1, que consisten en suprimir los espacios entre las palabras y escribir con mayúscula la primera letra de la palabra siguiente, por ejemplo, "managed object class" (clase de objeto gestionado) se convierte en "managedObjectClass". Algunos de los nombres se abrevian de la manera siguiente:

- ack acuse de recibo (*acknowledgement*)
- arg argumento (*argument*)
- id identificador (*identifier*)
- info información (*information*)
- sync sincronización (*synchronization*)

7.2 Correspondencia entre primitivas de CMISE y operaciones de CMIP

Cuadro 4 – Correspondencia entre primitivas de CMISE y operaciones de CMIP

Primitiva CMIS	Modo	ID de ligazón	Operación CMIP
pet./ind. M-CANCELACIÓN-OBTENCIÓN	Confirmado	No aplicable	m-Cancelación-Obtención-Confirmado
resp./conf. M-CANCELACIÓN-OBTENCIÓN	No aplicable	No aplicable	m-Cancelación-Obtención-Confirmado
pet./ind. M-INFORME-EVENTO	No confirmado	No aplicable	m-Informe-Evento
pet./ind. M-INFORME-EVENTO	Confirmado	No aplicable	m-Informe-Evento-Confirmado
resp./conf. M-INFORME-EVENTO	No aplicable	No aplicable	m-Informe-Evento-Confirmado
pet./ind. M-OBTENCIÓN	Confirmado	No aplicable	m-Obtención
resp./conf. M-OBTENCIÓN	No aplicable	Ausente	m-Obtención
resp./conf. M-OBTENCIÓN	No aplicable	Presente	m-Respuesta-Ligada
pet./ind. M-FIJACIÓN	No confirmado	No aplicable	m-Fijación
pet./ind. M-FIJACIÓN	Confirmado	No aplicable	m-Fijación-confirmado
resp./conf. M-FIJACIÓN	No aplicable	Ausente	m-Fijación-confirmado
resp./conf. M-FIJACIÓN	No aplicable	Presente	m-Respuesta-Ligada
pet./ind. M-ACCIÓN	No confirmado	No aplicable	m-Acción
pet./ind. M-ACCIÓN	Confirmado	No aplicable	m-Acción-confirmado
resp./conf. M-ACCIÓN	No aplicable	Ausente	m-Acción-confirmado
resp./conf. M-ACCIÓN	No aplicable	Presente	m-Respuesta-Ligada
pet./ind. M-CREACIÓN	Confirmado	No aplicable	m-Creación
resp./conf. M-CREACIÓN	No aplicable	No aplicable	m-Creación
pet./ind. M-SUPRESIÓN	Confirmado	No aplicable	m-Supresión
resp./conf. M-SUPRESIÓN	No aplicable	Ausente	m-Supresión
resp./conf. M-SUPRESIÓN	No aplicable	Presente	m-Respuesta-Ligada

NOTA – La correspondencia de las macros OPERATION (Operación) y ERROR con el ROSE es la definida en la Rec. X.219 del CCITT | ISO/CEI 9072-1.

7.3 Datos de usuario de ACSE

El protocolo de ACSE (véase la Rec. UIT-T X.227 | ISO/CEI 8650-1) se describe utilizando la ASN.1. La "información de usuario" se define utilizando el tipo de datos EXTERNAL.

7.3.1 Datos de usuario de A-ASOCIACIÓN

La codificación de la información de usuario CMIP que ha de pasarse a A-ASOCIACIÓN en el parámetro "información de usuario" se define de la manera siguiente:

CMIP-A-ASSOCIATE-Information {joint-iso-itu-t ms(9) cmip(1) modules(0) aAssociateUserInfo(1)}

DEFINITIONS ::= BEGIN

FunctionalUnits ::= BIT STRING {
multipleObjectSelection (0),
filter (1),
multipleReply (2),
extendedService (3),
cancelGet (4)
}

-- *Functional unit i is supported if and only if bit i is one*

-- *Information carried in user-information parameter of A-ASSOCIATE*

CMIPUserInfo ::= SEQUENCE {
protocolVersion [0] IMPLICIT ProtocolVersion DEFAULT { version1 },
functionalUnits [1] IMPLICIT FunctionalUnits DEFAULT {},
accessControl [2] EXTERNAL OPTIONAL,
userInfo [3] EXTERNAL OPTIONAL
}

ProtocolVersion ::= BIT STRING {
version1 (0),
version2 (1)
}

END

La codificación de otra "información de usuario" suministrada por el usuario del servicio CMISE no se define en esta Recomendación | Norma Internacional.

7.3.2 Datos de usuario de A-ABORTO

La codificación de la información de usuario CMIP que ha de pasarse a A-ABORTO en el parámetro "información de usuario" se define de la manera siguiente:

CMIP-A-ABORT-Information {joint-iso-itu-t ms(9) cmip(1) modules(0) aAbortUserInfo(2)}

DEFINITIONS ::= BEGIN

-- *Information carried in user-information parameter of A-ABORT*

CMIPAbortInfo ::= SEQUENCE {
abortSource [0] IMPLICIT CMIPAbortSource,
userInfo [1] EXTERNAL OPTIONAL
}

CMIPAbortSource ::= ENUMERATED {
cmiseServiceUser (0),
cmiseServiceProvider (1)
}

END

La codificación de otra "información de usuario" suministrada por el usuario del servicio CMISE no se define en esta Recomendación | Norma Internacional.

7.4 Unidades de datos CMIP

El protocolo se describe en términos de unidades de datos de protocolo común de información de gestión intercambiadas entre entidades pares del CMISE. Las PDU se especifican utilizando la ASN.1 y las macros externas OPERATION y ERROR del Protocolo de Operaciones a Distancia, definidas en la Rec. X.219 del CCITT | ISO/CEI 9072-1.

-- *Common Management Information Protocol (CMIP)*

CMIP-1 {joint-iso-itu-t ms(9) cmip(1) modules(0) protocol(3)}

DEFINITIONS ::= BEGIN

-- *Remote Operations definitions*

IMPORTS OPERATION, ERROR FROM Remote-Operation-Notation {joint-iso-ccitt remote-operations(4) notation(0)}

-- *Remote Operations Service definitions*

InvokeIDType FROM Remote-Operations-APDUs {joint-iso-ccitt remote-operations(4) apdus(1)}

-- *Directory Service definitions*

-- *This Recommendation | International Standard imports abstract syntax from CCITT Rec. X.501 (1988) |*

-- *ISO/IEC 9594-2:1990, Annex D to this Recommendation | International Standard provides an extract*

-- *from CCITT Rec. X.501 (1988) | ISO/IEC 9594-2:1990, sufficient to meet the needs of CMIP.*

DistinguishedName, RDNSSequence

FROM InformationFramework {joint-iso-ccitt ds(5) modules(1) informationFramework(1)};

-- *CMISE operations*

-- *in the following operations, the argument type is mandatory in the corresponding ROSE APDU*

-- *Action operations (M-ACTION)*

m-Action OPERATION

ARGUMENT ActionArgument

::= localValue : 6

m-Action-Confirmed OPERATION

ARGUMENT ActionArgument

RESULT ActionResult -- *this result is conditional; for conditions see 8.3.3.2.9 of ITU-T Rec. X.710 |*
-- *ISO/IEC 9595*

ERRORS { accessDenied, classInstanceConflict, complexityLimitation, invalidScope,
invalidArgumentValue, invalidFilter, noSuchAction, noSuchArgument, noSuchObjectClass,
noSuchObjectInstance, processingFailure, syncNotSupported }

LINKED { m-Linked-Reply }

::= localValue : 7

m-CancelGet OPERATION

ARGUMENT

getInvokeId InvokeIDType

RESULT

ERRORS { mistypedOperation, noSuchInvokeId, processingFailure }

::= localValue : 10

-- *Create operation (M-CREATE)*

m-Create OPERATION

ARGUMENT CreateArgument

RESULT CreateResult -- *this result is conditional; for conditions see 8.3.4.1.3 of ITU-T Rec. X.710 |*
-- *ISO/IEC 9595*

ERRORS { accessDenied, classInstanceConflict, duplicateManagedObjectInstance,
invalidAttributeValue, invalidObjectInstance, missingAttributeValue, noSuchAttribute,
noSuchObjectClass, noSuchObjectInstance, noSuchReferenceObject, processingFailure }

::= localValue : 8

-- *Delete operation (M-DELETE)*

m-Delete OPERATION

Argument DeleteArgument

RESULT DeleteResult -- *this result is conditional; for conditions see 8.3.5.2.8 of ITU-T Rec. X.710 |*
-- *ISO/IEC 9595*

ERRORS { accessDenied, classInstanceConflict, complexityLimitation, invalidFilter,
InvalidScope, noSuchObjectClass, noSuchObjectInstance, processingFailure, syncNotSupported }

```

LINKED { m-Linked-Reply }
 ::= localValue : 9

-- Event Reporting operations (M-EVENT-REPORT)

m-EventReport OPERATION
  ARGUMENT   EventReportArgument
  ::= localValue : 0

m-EventReport-Confirmed OPERATION
  ARGUMENT   EventReportArgument
  RESULT     EventReportResult -- optional
  ERRORS {   invalidArgumentValue, noSuchArgument, noSuchEventType, noSuchObjectClass,
             noSuchObjectInstance, processingFailure }
  ::= localValue : 1

-- Get operation (M-GET)

m-Get OPERATION
  ARGUMENT   GetArgument
  RESULT     GetResult -- this result is conditional; for conditions see 8.3.1.2.8 of ITU-T Rec. X.710 /
             -- ISO/IEC 9595
  ERRORS {   accessDenied, classInstanceConflict, complexityLimitation, getListError, invalidFilter, invalidScope,
             noSuchObjectClass, noSuchObjectInstance, operationCancelled, processingFailure, syncNotSupported
             }
  LINKED { m-Linked-Reply }
  ::= localValue : 3

-- Linked operation to M-GET, M-SET (Confirmed), M-ACTION (Confirmed), and M-DELETE

m-Linked-Reply OPERATION
  ARGUMENT   LinkedReplyArgument
  ::= localValue : 2

-- Set operations (M-SET)

m-Set OPERATION
  ARGUMENT   SetArgument
  ::= localValue : 4

m-Set-Confirmed OPERATION
  ARGUMENT   SetArgument
  RESULT     SetResult -- this result is conditional; for conditions see 8.3.2.2.9 of ITU-T Rec. X.710 /
             -- ISO/IEC 9595
  ERRORS {   accessDenied, classInstanceConflict, complexityLimitation, invalidFilter, invalidScope,
             noSuchObjectClass, noSuchObjectInstance, processingFailure, setListError, syncNotSupported }
  LINKED { m-Linked-Reply }
  ::= localValue : 5

-- CMIS error definitions
-- in the following errors, unless otherwise indicated, the parameter type is mandatory in the corresponding ROSE
-- APDU

accessDenied ERROR
 ::= localValue : 2

classInstanceConflict ERROR
  PARAMETER BaseManagedObjectId
  ::= localValue : 19

complexityLimitation ERROR
  PARAMETER ComplexityLimitation-- optional
  ::= localValue : 20

duplicateManagedObjectInstance ERROR
  PARAMETER ObjectInstance
  ::= localValue : 11

getListError ERROR
  PARAMETER GetListError
  ::= localValue : 7

```

invalidArgumentValue ERROR

PARAMETER InvalidArgumentValue
 ::= localValue : 15

invalidAttributeValue ERROR

PARAMETER Attribute
 ::= localValue : 6

invalidFilter ERROR

PARAMETER CMISFilter
 ::= localValue : 4

invalidObjectInstance ERROR

PARAMETER ObjectInstance
 ::= localValue : 17

invalidScope ERROR

PARAMETER Scope
 ::= localValue : 16

missingAttributeValue ERROR

PARAMETER SET OF AttributeId
 ::= localValue : 18

mistypedOperation ERROR

::= localValue : 21

noSuchAction ERROR

PARAMETER NoSuchAction
 ::= localValue : 9

noSuchArgument ERROR

PARAMETER NoSuchArgument
 ::= localValue : 14

noSuchAttribute ERROR

PARAMETER AttributeId
 ::= localValue : 5

noSuchEventType ERROR

PARAMETER NoSuchEventType
 ::= localValue : 13

noSuchInvokeId ERROR

PARAMETER InvokeIDType
 ::= localValue : 22

noSuchObjectClass ERROR

PARAMETER ObjectClass
 ::= localValue : 0

noSuchObjectInstance ERROR

PARAMETER ObjectInstance
 ::= localValue : 1

noSuchReferenceObject ERROR

PARAMETER ObjectInstance
 ::= localValue : 12

operationCancelled ERROR

::= localValue : 23

processingFailure ERROR

PARAMETER ProcessingFailure -- optional
 ::= localValue : 10

setListError ERROR

PARAMETER SetListError
 ::= localValue : 8

syncNotSupported ERROR

PARAMETER CMISync
 ::= localValue : 3

-- Supporting type definitions

AccessControl ::= EXTERNAL

ActionArgument ::= SEQUENCE {
COMPONENTS OF **BaseManagedObjectId,**
accessControl **[5] AccessControl OPTIONAL,**
synchronization **[6] IMPLICIT CMISSync DEFAULT bestEffort,**
scope **[7] Scope DEFAULT namedNumbers : baseObject,**
filter **CMISFilter DEFAULT and : {},**
actionInfo **[12] IMPLICIT ActionInfo**
}

ActionError ::= SEQUENCE {
managedObjectClass **ObjectClass OPTIONAL,**
managedObjectInstance **ObjectInstance OPTIONAL,**
currentTime **[5] IMPLICIT GeneralizedTime OPTIONAL,**
actionErrorInfo **[6] ActionErrorInfo**
}

ActionErrorInfo ::= SEQUENCE {
errorStatus **ENUMERATED {** **accessDenied** **(2),**
 noSuchAction **(9),**
 noSuchArgument **(14),**
 invalidArgumentValue **(15) },**
errorInfo **CHOICE {**
 actionType **ActionTypeId,**
 actionArgument **[0] NoSuchArgument,**
 argumentValue **[1] InvalidArgumentValue**
}}

ActionInfo ::= SEQUENCE {
actionType **ActionTypeId,**
actionInfoArg **[4] ANY DEFINED BY actionType OPTIONAL**
}

ActionReply ::= SEQUENCE {
actionType **ActionTypeId,**
actionReplyInfo **[4] ANY DEFINED BY actionType**
}

ActionResult ::= SEQUENCE {
managedObjectClass **ObjectClass OPTIONAL,**
managedObjectInstance **ObjectInstance OPTIONAL,**
currentTime **[5] IMPLICIT GeneralizedTime OPTIONAL,**
actionReply **[6] IMPLICIT ActionReply OPTIONAL**
}

ActionTypeId ::= CHOICE {
globalForm **[2] IMPLICIT OBJECT IDENTIFIER,**
localForm **[3] IMPLICIT INTEGER**
}

-- This Recommendation / International Standard does not allocate any values for localForm. Where this alternative is
 -- used, the permissible values for the integers and their meanings shall be defined as part of the application context in
 -- which they are used

Attribute ::= SEQUENCE {
attributeId **AttributeId,**
attributeValue **ANY DEFINED BY attributeId**
}

AttributeError ::= SEQUENCE {
errorStatus **ENUMERATED {** **accessDenied** **(2),**
 noSuchAttribute **(5),**
 invalidAttributeValue(6),
 invalidOperation **(24),**
 invalidOperator **(25) },**
modifyOperator **[2] IMPLICIT ModifyOperator OPTIONAL, -- present for invalidOperator & invalidOperation**
attributeId **AttributeId,**
attributeValue **ANY DEFINED BY attributeId OPTIONAL -- absent for setToDefault**
}

AttributeId ::= CHOICE {

globalForm [0] IMPLICIT OBJECT IDENTIFIER,
localForm [1] IMPLICIT INTEGER
}

-- This Recommendation / International Standard does not allocate any values for localForm. Where this alternative is
 -- used, the permissible values for the integers and their meanings shall be defined as part of the application context in
 -- which they are used

AttributeIdError ::= SEQUENCE {

errorStatus ENUMERATED { accessDenied (2),
noSuchAttribute (5) },
attributeId AttributeId
}

BaseManagedObjectId ::= SEQUENCE {

baseManagedObjectClass ObjectClass,
baseManagedObjectInstance ObjectInstance
}

CMISFilter ::= CHOICE {

item [8] FilterItem,
and [9] IMPLICIT SET OF CMISFilter,
or [10] IMPLICIT SET OF CMISFilter,
not [11] CMISFilter
}

CMISSync ::= ENUMERATED {

bestEffort (0),
atomic (1) }

ComplexityLimitation ::= SET {

scope [0] Scope OPTIONAL,
filter [1] CMISFilter OPTIONAL,
sync [2] CMISSync OPTIONAL
}

CreateArgument ::= SEQUENCE {

managedObjectClass ObjectClass,
managedOrSuperiorObjectInstance CHOICE {
managedObjectInstance ObjectInstance,
superiorObjectInstance [8] ObjectInstance } OPTIONAL,
accessControl [5] AccessControl OPTIONAL,
referenceObjectInstance [6] ObjectInstance OPTIONAL,
attributeList [7] IMPLICIT SET OF Attribute OPTIONAL
}

CreateResult ::= SEQUENCE {

managedObjectClass ObjectClass OPTIONAL,
managedObjectInstance ObjectInstance OPTIONAL, -- shall be returned if omitted from CreateArgument
currentTime [5] IMPLICIT GeneralizedTime OPTIONAL,
attributeList [6] IMPLICIT SET OF Attribute OPTIONAL
}

DeleteArgument ::= SEQUENCE {

COMPONENTS OF BaseManagedObjectId,
accessControl [5] AccessControl OPTIONAL,
synchronization [6] IMPLICIT CMISSync DEFAULT bestEffort,
scope [7] Scope DEFAULT namedNumbers : baseObject,
filter CMISFilter DEFAULT and : {}
}

DeleteError ::= SEQUENCE {

managedObjectClass ObjectClass OPTIONAL,
managedObjectInstance ObjectInstance OPTIONAL,
currentTime [5] IMPLICIT GeneralizedTime OPTIONAL,
deleteErrorInfo [6] ENUMERATED { accessDenied (2)
}}

DeleteResult ::= SEQUENCE {

managedObjectClass ObjectClass OPTIONAL,
managedObjectInstance ObjectInstance OPTIONAL,
currentTime [5] IMPLICIT GeneralizedTime OPTIONAL
}


```

EventReply ::= SEQUENCE {
    eventType      EventTypeId,
    eventReplyInfo [8] ANY DEFINED BY eventType OPTIONAL
}

```

```

EventReportArgument ::= SEQUENCE {
    managedObjectClass      ObjectClass,
    managedObjectInstance   ObjectInstance,
    eventTime                [5] IMPLICIT GeneralizedTime OPTIONAL,
    eventType                EventTypeId,
    eventInfo                [8] ANY DEFINED BY eventType OPTIONAL
}

```

```

EventReportResult ::= SEQUENCE {
    managedObjectClass      ObjectClass OPTIONAL,
    managedObjectInstance   ObjectInstance OPTIONAL,
    currentTime            [5] IMPLICIT GeneralizedTime OPTIONAL,
    eventReply              EventReply OPTIONAL
}

```

```

EventTypeId ::= CHOICE {
    globalForm      [6] IMPLICIT OBJECT IDENTIFIER,
    localForm      [7] IMPLICIT INTEGER
}

```

-- This Recommendation / International Standard does not allocate any values for localForm. Where this alternative is used, the permissible values for the integers and their meanings shall be defined as part of the application context in which they are used

```

FilterItem ::= CHOICE {
    equality          [0] IMPLICIT Attribute,
    substrings       [1] IMPLICIT SEQUENCE OF CHOICE {
        initialString [0] IMPLICIT SEQUENCE {
            attributeIdAttributeId,
            string      ANY DEFINED BY attributeId },
        anyString      [1] IMPLICIT SEQUENCE {
            attributeIdAttributeId,
            string      ANY DEFINED BY attributeId },
        finalString    [2] IMPLICIT SEQUENCE {
            attributeIdAttributeId,
            string      ANY DEFINED BY attributeId },
    greaterOrEqual   [2] IMPLICIT Attribute, -- asserted value ≥ attribute value
    lessOrEqual      [3] IMPLICIT Attribute, -- asserted value ≥ attribute value
    present          [4] AttributeId,
    subsetOf         [5] IMPLICIT Attribute, -- asserted value is a subset of attribute value
    supersetOf       [6] IMPLICIT Attribute, -- asserted value is a superset of attribute value
    nonNullSetIntersection [7] IMPLICIT Attribute
}

```

```

GetArgument ::= SEQUENCE {
    COMPONENTS OF      BaseManagedObjectId,
    accessControl      [5] AccessControl OPTIONAL,
    synchronization    [6] IMPLICIT CMISync DEFAULT bestEffort,
    scope              [7] Scope DEFAULT namedNumbers : baseObject,
    filter              CMISFilter DEFAULT and : {},
    attributeIdList    [12] IMPLICIT SET OF AttributeId OPTIONAL
}

```

```

GetInfoStatus ::= CHOICE {
    attributeIdError [0] IMPLICIT AttributeIdError,
    attribute        [1] IMPLICIT Attribute
}

```

```

GetListError ::= SEQUENCE {
    managedObjectClass      ObjectClass OPTIONAL,
    managedObjectInstance   ObjectInstance OPTIONAL,
    currentTime            [5] IMPLICIT GeneralizedTime OPTIONAL,
    getInfoList             [6] IMPLICIT SET OF GetInfoStatus
}

```

```

GetResult ::= SEQUENCE {
    managedObjectClass ObjectClass OPTIONAL,
    managedObjectInstance ObjectInstance OPTIONAL,
    currentTime [5] IMPLICIT GeneralizedTime OPTIONAL,
    attributeList [6] IMPLICIT SET OF Attribute OPTIONAL
}

```

```

InvalidArgumentValue ::= CHOICE {
    actionValue [0] IMPLICIT ActionInfo,
    eventValue [1] IMPLICIT SEQUENCE {
        eventType EventTypeId,
        eventInfo [8] ANY DEFINED BY eventType OPTIONAL
    }
}

```

```

LinkedReplyArgument ::= CHOICE {
    getResult [0] IMPLICIT GetResult,
    getListError [1] IMPLICIT GetListError,
    setResult [2] IMPLICIT SetResult,
    setListError [3] IMPLICIT SetListError,
    actionResult [4] IMPLICIT ActionResult,
    processingFailure [5] IMPLICIT ProcessingFailure,
    deleteResult [6] IMPLICIT DeleteResult,
    actionError [7] IMPLICIT ActionError,
    deleteError [8] IMPLICIT DeleteError
}

```

```

ModifyOperator ::= INTEGER {
    replace (0),
    addValues (1),
    removeValues (2),
    setToDefault (3)
}

```

```

NoSuchAction ::= SEQUENCE {
    managedObjectClass ObjectClass,
    actionType ActionTypeId
}

```

```

NoSuchArgument ::= CHOICE {
    actionId [0] IMPLICIT SEQUENCE {
        managedObjectClass ObjectClass OPTIONAL,
        actionType ActionTypeId
    },
    eventId [1] IMPLICIT SEQUENCE {
        managedObjectClass ObjectClass OPTIONAL,
        eventType EventTypeId
    }
}

```

```

NoSuchEventType ::= SEQUENCE {
    managedObjectClass ObjectClass,
    eventType EventTypeId
}

```

```

ObjectClass ::= CHOICE {
    globalForm [0] IMPLICIT OBJECT IDENTIFIER,
    localForm [1] IMPLICIT INTEGER
}

```

-- This Recommendation / International Standard does not allocate any values for localForm. Where this alternative is used, the permissible values or the integers and their meanings shall be defined as part of the application context in which they are used

```

ObjectInstance ::= CHOICE {
    distinguishedName [2] IMPLICIT DistinguishedName,
    nonSpecificForm [3] IMPLICIT OCTET STRING,
    localDistinguishedName [4] IMPLICIT RDNSSequence
}

```

-- localDistinguishedName is that portion of the distinguished name that is necessary to unambiguously identify the managed object within the context of communication between the open systems

```

ProcessingFailure ::= SEQUENCE {
    managedObjectClass ObjectClass,
    managedObjectInstance ObjectInstance OPTIONAL,
    specificErrorInfo [5] SpecificErrorInfo
}

```

```

Scope ::= CHOICE { namedNumbers INTEGER { baseObject (0),
                                           firstLevelOnly (1),
                                           wholeSubtree (2) },
  individualLevels [1] IMPLICIT INTEGER, -- POSITIVE integer indicates the level to be selected
  baseToNthLevel [2] IMPLICIT INTEGER, -- POSITIVE integer N indicates that the range of levels
                                           -- (0 – N) is to be selected
-- with individualLevels and baseToNthLevel, a value of 0 has the same semantics as baseObject
-- with individualLevels, a value of 1 has the same semantics as firstLevelOnly

SetArgument ::= SEQUENCE {
  COMPONENTS OF BaseManagedObjectId,
  accessControl [5] AccessControl OPTIONAL,
  synchronization [6] IMPLICIT CMISync DEFAULT bestEffort,
  scope [7] Scope DEFAULT namedNumbers : baseObject,
  filter CMISFilter DEFAULT and : {},
  modificationList [12] IMPLICIT SET OF SEQUENCE {
    modifyOperator [2] IMPLICIT ModifyOperator DEFAULT replace,
    attributeId AttributeId,
    attributeValue ANY DEFINED BY attributeId OPTIONAL -- absent for setToDefault
  }
}

SetInfoStatus ::= CHOICE {
  attributeError [0] IMPLICIT AttributeError,
  attribute [1] IMPLICIT Attribute
}

SetListError ::= SEQUENCE {
  managedObjectClass ObjectClass OPTIONAL,
  managedObjectInstance ObjectInstance OPTIONAL,
  currentTime [5] IMPLICIT GeneralizedTime OPTIONAL,
  setInfoList [6] IMPLICIT SET OF SetInfoStatus
}

SetResult ::= SEQUENCE {
  managedObjectClass ObjectClass OPTIONAL,
  managedObjectInstance ObjectInstance OPTIONAL,
  currentTime [5] IMPLICIT GeneralizedTime OPTIONAL,
  attributeList [6] IMPLICIT SET OF Attribute OPTIONAL
}

SpecificErrorInfo ::= SEQUENCE {
  errorId OBJECT IDENTIFIER,
  errorInfo ANY DEFINED BY errorId
}

END -- End of CMIP syntax definitions

```

7.5 Definición de sintaxis abstracta para CMIP

Esta Recomendación | Norma Internacional asigna el valor de identificador de objeto de ASN.1

{joint-iso-itu-t ms(9) cmip(1) cmip-pci(1) abstractSyntax(4)}

como un nombre de sintaxis abstracta para el conjunto de valores de datos de presentación, cada uno de los cuales es un valor del tipo ASN.1

Remote-Operations-APDUs.ROSEapdus

definido en la Rec. X.229 del CCITT | ISO/CEI 9072-2 con el componente de argumento relleno de acuerdo con las definiciones en CMIP-1, o un valor de uno de los tipos ASN.1:

- **CMIP-A-ASSOCIATE-Information.CMIPUserInfo;**
- **CMIP-A-ABORT-Information.CMIPAbortInfo.**

El correspondiente valor de descriptor de objeto será

“CMIP-PCI”.

Esta sintaxis abstracta se define para incluir todos los tipos de datos resueltos por las producciones ANY DEFINED BY X, donde X es de tipo OBJECT IDENTIFIER.

Los valores de identificador de objeto y de descriptor de objeto

{joint-iso-ccitt asn1(1) basic-encoding(1)} and “Basic Encoding of single ASN.1 type”

(asignados a un objeto en la Rec. X.209 del CCITT | ISO/CEI 8825) pueden utilizarse como un nombre de sintaxis de transferencia con esta sintaxis abstracta.

7.5.1 Reglas de extensibilidad

7.5.1.1 Cuando se procesa información A-ASOCIACIÓN de CMIP entrante, la CMIPM:

- ignorará todos los valores rotulados ("tagged") que no están definidos en la sintaxis abstracta de esta Recomendación | Norma Internacional; e
- ignorará todas las asignaciones de nombre de bit desconocidas dentro de una BIT STRING.

7.5.1.2 El nombre de sintaxis abstracta puede utilizarse cuando se modifican los valores de datos de presentación para incluir:

- operaciones de gestión de un nuevo sistema;
- nuevos elementos rotulados dentro de un SET o SEQUENCE;
- nuevas asignaciones de nombre de bit dentro de una BIT STRING;
- nuevos números denominados para un INTEGER; y
- nuevas enumeraciones denominadas dentro de un ENUMERATED.

8 Conformidad

Un sistema que afirma implementar los procedimientos especificados en esta Recomendación | Norma Internacional cumplirá los requisitos indicados en 8.1 y 8.2.

8.1 Requisitos estáticos

Una implementación a la que se exige el cumplimiento de esta Recomendación | Norma Internacional debe ir acompañada de un PICS de conformidad con el formulario de PICS especificado por la Rec. UIT-T X.712 | ISO/CEI 9596-2. En particular, se establecerá que la implementación:

- 1) soportará el protocolo, requerido para proporcionar todos los servicios CMIS definidos en la unidad funcional kernel (o de núcleo), en ambas funciones de invocador y realizador; o
- 2) soportará el protocolo, requerido para proporcionar alguno de los servicios CMIS definidos en la unidad funcional kernel, en las funciones de invocador y/o realizador, pero no todos los protocolos en ambas funciones para todos los servicios CMIS.

La implementación:

- a) soportará los elementos de procedimiento especificados en la cláusula 6 de la presente Recomendación | Norma Internacional para aquellos servicios en la función para que se afirma la conformidad;
- b) soportará la sintaxis abstracta asociada con las unidades de datos de protocolo que son necesarias para cursar las peticiones y/o respuestas para aquellos servicios en la función para que se afirma la conformidad;
- c) soportará la sintaxis de transferencia derivada de las reglas de codificación especificadas en la Rec. X.209 del CCITT | ISO/CEI 8825 y denominada:

{joint-iso-ccitt asn1(1) basic-encoding(1)},

a los efectos de generar e interpretar información de unidades PDU de CMIP definida por la sintaxis abstracta

“CMIP-PCF”,

para los elementos de protocolo a los que se reclama conformidad:

- d) soportará el protocolo ACSE definido en la Rec. UIT-T X.227 | ISO/CEI 8650-1, para establecer y liberar una asociación;
- e) soportará las reglas especificadas en el anexo A en cualquier contexto de aplicación que incluya CMISE como uno de los ASE;

- f) soportará la clase 3 de protocolo ROSE definida en la Rec. X.229 del CCITT | ISO/CEI 9072-2;
- g) soportará los elementos de protocolo requeridos para proporcionar la unidad funcional de respuestas múltiples si se soporta la unidad funcional de selección de objetos múltiples;
- h) soportará los elementos de protocolo requeridos para proporcionar el servicio M-OBTENCIÓN si se soporta la unidad funcional cancelar obtención.

8.2 Requisitos dinámicos

El sistema:

- a) soportará los procedimientos para cada uno de los servicios CMIS en la función para el que se afirma la conformidad;
- b) verificará los parámetros de seguridad facultativos definidos en las PDU de CMIP, cuando los utilice;
- c) cuando se soporta la unidad funcional de servicio ampliada, soportará el protocolo de presentación definido en la Rec. UIT-T X.226 | ISO/CEI 8823-1, según lo requiera el contexto de aplicación;
- d) cuando se proporcione determinación de ámbito, soportará los elementos de procedimiento para la unidad funcional de respuestas múltiples.

Anexo A

Reglas de asociación para el CMISE

(Este anexo no forma parte de esta Recomendación | Norma Internacional)

A.1 ACSE, requisitos de sesión y presentación

A.1.1 El CMISE requiere la unidad funcional de presentación kernel definida en la Rec. UIT-T X.216 | ISO/CEI 8822.

A.1.2 El CMISE requiere las unidades funcionales kernel y de sesión dúplex definidas en la Rec. UIT-T X.215 | ISO/CEI 8326.

A.1.3 El CMISE requiere el modo normal de ACSE y los servicios de presentación definidos en las Recomendaciones UIT-T X.227 | ISO/CEI 8650-1 y UIT-T X.216 | ISO/CEI 8822.

A.2 Reglas de iniciación de asociación

A.2.1 Petición

El usuario del servicio CMISE que inicia el establecimiento de asociación proporcionará la información de usuario A-ASOCIACIÓN definida en la Rec. UIT-T X.710 | ISO/CEI 9595. La información de usuario CMIP se proporcionará a la CMIPM, la cual:

- a) construirá CMIPUserInfo (información de usuario CMIP) a partir de la información suministrada;
- b) fijará el parámetro de versión de protocolo de la información de usuario CMIP fijando el bit correspondiente a cada versión soportada;
- c) incluirá la información de usuario CMIP como un EXTERNAL separado en el parámetro información de usuario de la primitiva petición A-ASOCIACIÓN;
- d) esperará a que la información de usuario específica del CMIS sea devuelta en la primitiva confirmación A-ASOCIACIÓN.

A.2.2 Indicación

Al recibir una primitiva indicación A-ASOCIACIÓN, se proporcionará el parámetro información de usuario CMIP a la CMIPM, la cual:

- a) comprobará que se soporta por lo menos una de las versiones de protocolo propuestas;
- b) verificará que el parámetro de control de acceso facultativo es válido;
- c) si fracasa cualquiera de las comprobaciones, se rechazará la asociación fijando el motivo del parámetro de fallo de la primitiva respuesta A-ASOCIACIÓN a "rechazado por respondedor (permanente)". La asociación no se establece y esa instancia (o manifestación) de la CMIPM dejará de existir;
- d) si estas comprobaciones tienen éxito, la información siguiente, si está presente, se facilitará al usuario de servicio CMISE: unidades funcionales soportadas por el proveedor del servicio CMISE, control de acceso e información de usuario. La CMIPM esperará la respuesta del usuario del servicio CMISE.

A.2.3 Respuesta

La primitiva respuesta A-ASOCIACIÓN que indica "aceptado" o "rechazado", y que, si es aceptada, incluye las unidades funcionales y los parámetros control de acceso e información de usuario, se proporcionará a la CMIPM, la cual:

- a) construirá la CMIPUserInfo requerida para la respuesta. La información de usuario del CMIP incluirá el parámetro de versión que indica todas las versiones de CMIP que son soportadas;
- b) incluirá la información de usuario del CMIP como un EXTERNAL separado en el parámetro de información de usuario de la primitiva respuesta A-ASOCIACIÓN;
- c) si la respuesta de asociación indica "aceptada", la versión de protocolo acordada es la correspondiente al número más alto soportado por ambas CMIPM. La CMIPM estará preparada entonces para aceptar primitivas indicación CMISE;
- d) si la respuesta de asociación indica "rechazada", la instancia de la CMIPM dejará de existir.

A.2.4 Confirmación

Al recibir la primitiva confirmación A-ASOCIACIÓN, se proporcionará el parámetro información de usuario del CMIP a la CMIPM, la cual:

- a) si la confirmación de asociación indica éxito, la asociación se establece y se proporcionan al iniciador de la asociación las unidades funcionales, y los parámetros control de acceso e información de usuario. Las unidades funcionales acordadas corresponden a aquellas que ambos usuarios del servicio CMISE han indicado que soportan y la versión de protocolo es el número de versión más alto soportado por ambas CMIPM;
- b) si la confirmación de asociación indica fallo, no se establece la asociación y la instancia de la CMIPM dejará de existir.

A.3 Reglas de liberación de asociación

Cualquiera de los dos usuarios del servicio CMISE puede iniciar una liberación de asociación.

A.3.1 Petición

Al recibir una petición de liberación de asociación, se proporcionarán los parámetros de A-LIBERACIÓN necesarios a la CMIPM, la cual dejará de aceptar peticiones de servicio y esperará confirmación de la liberación de la asociación.

A.3.2 Indicación

Al recibir una primitiva indicación A-LIBERACIÓN, se proporcionarán los parámetros de indicación A-LIBERACIÓN necesarios a la CMIPM que responde, la cual esperará la respuesta de la liberación de asociación.

A.3.3 Respuesta

Al recibir una respuesta de liberación de asociación del usuario del servicio CMISE que responde, se proporcionarán los parámetros de respuesta A-LIBERACIÓN necesarios a la CMIPM que responde. Después de esto, esta instancia de la CMIPM dejará de existir.

A.3.4 Confirmación

Al recibir una primitiva confirmación A-LIBERACIÓN, se proporcionarán los parámetros de confirmación de A-LIBERACIÓN necesarios a la CMIPM iniciadora. Después de esto, esta instancia de la CMIPM dejará de existir.

A.4 Reglas de aborto de asociación

Cualquiera de los dos usuarios del servicio CMISE puede iniciar una terminación abrupta de la asociación.

En función de la información local, si los servicios subyacentes no disponen de la capacidad de transportar información de usuario ilimitada por A-ABORTO, el parámetro CMIPAbortInfo puede no estar incluido en las primitivas de servicio A-ABORTO.

A.4.1 Petición A-ABORTO

Al recibir una petición de aborto de asociación, se proporcionarán los parámetros de petición A-ABORTO necesarios, que incluyen la información de usuario A-ABORTO definida en la Rec. UIT-T X.710 | ISO/CEI 9595, a la CMIPM, la cual:

- a) construirá CMIPAbortInfo (información de aborto del CMIP) a partir de la información proporcionada;
- b) fijará el parámetro fuente de aborto de la información de aborto del CMIP a usuario del servicio CMISE;
- c) incluirá información de aborto del CMIP como un campo separado en el parámetro Información de usuario de la primitiva petición A-ABORTO;
- d) después de esto, esta instancia de la CMIPM dejará de existir.

A.4.2 Indicación A-ABORTO

Al recibir una primitiva indicación A-ABORTO, se proporcionarán a la CMIPM los parámetros indicación A-ABORTO necesarios que incluyen CMIPAbortInfo (información de aborto del CMIP). Después de esto, esta instancia de la CMIPM dejará de existir.

A.4.3 Indicación A-P-ABORTO

Al recibir una primitiva indicación A-P-ABORTO, se proporcionarán a la CMIPM los parámetros de indicación A-P-ABORTO necesarios. Después de esto, esta instancia de la CMIPM dejará de existir.

A.4.4 Error de protocolo CMIP

Al detectar un error de protocolo, la CMIPM:

- a) construirá la información de aborto del CMIP con el parámetro fuente de aborto fijado a proveedor del servicio CMISE;
- b) indicará al usuario del servicio CMISE que se ha producido un error de protocolo;
- c) incluirá la información de aborto del CMIP como un campo separado en el parámetro información de usuario de la primitiva petición A-ABORTO;
- d) después de esto, esta instancia de la CMIPM dejará de existir.

Anexo B

Sintaxis ASN.1 ampliada

(Este anexo no es parte de esta Recomendación | Norma Internacional)

En este anexo se describe cómo las macros OPERATION y ERROR de la Rec. X.219 del CCITT | ISO/CEI 9072-1 se amplían a tipos y subtipos de datos ASN.1.

Si existen incoherencias entre estas definiciones y las de la cláusula 7, tendrán precedencia estas últimas.

-- *Common Management Information Protocol (CMIP)*

CMIP-1 {joint-iso-itu-t ms(9) cmip(1) modules(0) protocol(3)}

DEFINITIONS ::= BEGIN

-- *Remote Operations definitions*

IMPORTS OPERATION, ERROR FROM Remote-Operation-Notation {joint-iso-ccitt remote-operations(4) notation(0)}

-- *Directory Service definitions*

DistinguishedName, RDNSequence

FROM InformationFramework {joint-iso-ccitt ds(5) modules(1) informationFramework(1)};

-- *CMISE operations*

ROSEapdu ::= CHOICE {

roiv-apdu [1] IMPLICIT ROIVapdu,
 rors-apdu [2] IMPLICIT RORSapdu,
 roer-apdu [3] IMPLICIT ROERapdu,
 rorj-apdu [4] IMPLICIT RORJapdu
}

ROIVapdu ::= SEQUENCE {

invokeID InvokeIDType,
 linked-ID [0] IMPLICIT InvokeIDType OPTIONAL,
 operation-value OPERATION,
 argument ANY DEFINED BY operation-value OPTIONAL
}

RORSapdu ::= SEQUENCE {

invokeID InvokeIDType,
 SEQUENCE { **operation-value** OPERATION,
 result ANY DEFINED BY operation-value } OPTIONAL
}

ROERapdu ::= SEQUENCE {

invokeID InvokeIDType,
 error-value ERROR,
 parameter ANY DEFINED BY error-value OPTIONAL
}

RORJapdu ::= SEQUENCE {

invokeID CHOICE {InvokeIDType,
 NULL },
 problem CHOICE { [0] IMPLICIT GeneralProblem,
 [1] IMPLICIT InvokeProblem,
 [2] IMPLICIT ReturnResultProblem,
 [3] IMPLICIT ReturnErrorProblem }
}

InvokeIDType ::= INTEGER

-- *The use of the GeneralProblem, ReturnResultProblem, and ReturnErrorProblem codes are a local issue.*

GeneralProblem ::= INTEGER {

unrecognisedAPDU (0), -- ROSE-provider detected
 mistypedAPDU (1),
 badlyStructuredAPDU (2)
}

```

InvokeProblem ::= INTEGER {
    duplicateInvocation      (0), -- ROSE-user detected
    unrecognisedOperation    (1),
    mistypedArgument        (2),
    resourceLimitation      (3),
    initiatorReleasing      (4),
    unrecognisedLinkedID    (5),
    linkedResponseUnexpected (6),
    unexpectedChildOperation (7)
}

```

```

ReturnResultProblem ::= INTEGER {
    unrecognisedInvocation    (0), -- ROSE-user detected
    resultResponseUnexpected  (1),
    mistypedResult          (2)
}

```

```

ReturnErrorProblem ::= INTEGER {
    unrecognisedInvocation    (0), -- ROSE-user detected
    errorResponseUnexpected   (1),
    unrecognisedError        (2),
    unexpectedError          (3),
    mistypedParameter        (4)
}

```

-- This part of the ASN.1 specification provides a definition of the InvokeProblem subtype used by CMIP.

```

InvokeProblem-CMIPUser ::= InvokeProblem (
    duplicateInvocation      |
    unrecognisedOperation    |
    mistypedArgument        |
    resourceLimitation
)

```

-- This part of the ASN.1 specification provides a definition of ROIVapdu and RORSapdu subtypes used by CMIP.

-- The subtypes of the ROIVapdu define the allowed values of the operation-value and argument defined by that operation-value for all CMIP notifications and operations. The subtypes of the RORSapdu define the allowed values of the operation-value and result defined by that operation-value for all CMIP notifications and operations.

m-Action OPERATION ::= localValue : 6

```

ROIV-m-Action ::= ROIVapdu (WITH COMPONENTS {
    invokeID          PRESENT,
    linked-ID         ABSENT,
    operation-value   (m-Action),
    argument          (INCLUDES ActionArgument) } )

```

m-Action-Confirmed OPERATION ::= localValue : 7

```

ROIV-m-Action-Confirmed ::= ROIVapdu (WITH COMPONENTS {
    invokeID          PRESENT,
    linked-ID         ABSENT,
    operation-value   (m-Action-Confirmed),
    argument          (INCLUDES ActionArgument) } )

```

```

RORS-m-Action-Confirmed ::= RORSapdu (WITH COMPONENTS {
    ...,
    invokeID          PRESENT,
    -- result sequence -- (WITH COMPONENTS
    { operation-value   (m-Action-Confirmed),
    result             (INCLUDES ActionResult) } )
    -- required only if there is a single reply to the ROIV-m-Action-Confirmed ROIVapdu and data is to be returned in
    -- the RORSapdu
})

```

```

m-Cancel-Get OPERATION ::= localValue : 10

ROIV-m-Cancel-Get ::= ROIVapdu (WITH COMPONENTS {
  invokeID PRESENT,
  linked-ID ABSENT,
  operation-value (m-Cancel-Get),
  argument (INCLUDES InvokeIDType)
})

RORS-m-Cancel-Get ::= RORSapdu (WITH COMPONENTS {
  invokeID PRESENT,
  -- There is no result sequence for RORS-m-Cancel-Get
})

m-Create OPERATION ::= localValue : 8

ROIV-m-Create ::= ROIVapdu (WITH COMPONENTS {
  invokeID PRESENT,
  linked-ID ABSENT,
  operation-value (m-Create),
  argument (INCLUDES CreateArgument) })

RORS-m-Create ::= RORSapdu (WITH COMPONENTS {
  ...,
  invokeID PRESENT,
  -- result sequence -- (WITH COMPONENTS
  { operation-value (m-Create),
  result (INCLUDES CreateResult) })
})

m-Delete OPERATION ::= localValue : 9

ROIV-m-Delete ::= ROIVapdu (WITH COMPONENTS {
  invokeID PRESENT,
  linked-ID ABSENT,
  operation-value (m-Delete),
  argument (INCLUDES DeleteArgument) })

RORS-m-Delete ::= RORSapdu (WITH COMPONENTS {
  ...,
  invokeID PRESENT,
  -- result sequence -- (WITH COMPONENTS
  { operation-value (m-Delete),
  result (INCLUDES DeleteResult) })
  -- required only if there is a single reply to the ROIV-m-DeleteROIVapdu and data is to be returned in the RORSapdu
})

m-EventReport OPERATION ::= localValue : 0

ROIV-m-EventReport ::= ROIVapdu (WITH COMPONENTS {
  invokeID PRESENT,
  linked-ID ABSENT,
  operation-value (m-EventReport),
  argument (INCLUDES EventReportArgument) })

m-EventReport-Confirmed OPERATION ::= localValue : 1

ROIV-m-EventReport-Confirmed ::= ROIVapdu (WITH COMPONENTS {
  invokeID PRESENT,
  linked-ID ABSENT,
  operation-value (m-EventReport-Confirmed),
  argument (INCLUDES EventReportArgument) })

RORS-m-EventReport-Confirmed ::= RORSapdu (WITH COMPONENTS {
  ...,
  invokeID PRESENT,
  -- result sequence -- (WITH COMPONENTS
  { operation-value (m-EventReport-Confirmed),
  result (INCLUDES EventReportResult) })
  -- required only if data is to be returned in the RORSapdu
})

```

m-Get OPERATION ::= localValue : 3

ROIV-m-Get ::= ROIVapdu (WITH COMPONENTS {
invokeID PRESENT,
linked-ID ABSENT,
operation-value (m-Get),
argument (INCLUDES GetArgument) })

RORS-m-Get ::= RORSapdu (WITH COMPONENTS {
... ,
invokeID PRESENT,
-- result sequence -- (WITH COMPONENTS
{ operation-value (m-Get),
result (INCLUDES GetResult) })
-- required only if there is a single reply to the ROIV-m-Get ROIVapdu
})

m-Linked-Reply OPERATION ::= localValue : 2

ROIV-m-Linked-Reply ::= ROIVapdu (WITH COMPONENTS {
invokeID PRESENT,
linked-ID PRESENT,
operation-value (m-Linked-Reply),
argument (INCLUDES LinkedReplyArgument) })

-- This part of the ASN.1 specification provides a definition of ROIV-m-Linked-Reply subtypes used by CMIP. The
 -- subtypes of the ROIV-m-Linked-Reply ROIVapdu define the allowed values of the argument defined by the
 -- operation-value for the specific CMIP linked reply operations.

ROIV-m-Linked-Reply-Action ::= ROIV-m-Linked-Reply (WITH COMPONENTS {
invokeID PRESENT,
linked-ID PRESENT,
operation-value (m-Linked-Reply),
argument (INCLUDES LinkedReplyArgument (WITH COMPONENTS {
getResult ABSENT,
getListError ABSENT,
setResult ABSENT,
setListError ABSENT,
actionResult PRESENT,
processingFailure PRESENT,
deleteResult ABSENT,
actionError PRESENT,
deleteError ABSENT })
})

ROIV-m-Linked-Reply-Delete ::= ROIV-m-Linked-Reply (WITH COMPONENTS {
invokeID PRESENT,
linked-ID PRESENT,
operation-value (m-Linked-Reply),
argument (INCLUDES LinkedReplyArgument (WITH COMPONENTS {
getResult ABSENT,
getListError ABSENT,
setResult ABSENT,
setListError ABSENT,
actionResult ABSENT,
processingFailure PRESENT,
deleteResult PRESENT,
actionError ABSENT,
deleteError PRESENT })
})

ROIV-m-Linked-Reply-Get ::= ROIV-m-Linked-Reply (WITH COMPONENTS {
invokeID PRESENT,
linked-ID PRESENT,
operation-value (m-Linked-Reply),
argument (INCLUDES LinkedReplyArgument (WITH COMPONENTS {
getResult PRESENT,
getListError PRESENT,
setResult ABSENT,
setListError ABSENT,
})

```

        actionResult      ABSENT,
        processingFailure  PRESENT,
        deleteResult      ABSENT,
        actionError       ABSENT,
        deleteError       ABSENT } )
    ) )
ROIV-m-Linked-Reply-Set ::= ROIV-m-Linked-Reply (WITH COMPONENTS {
    invokeID      PRESENT,
    linked-ID     PRESENT,
    operation-value (m-Linked-Reply),
    argument      (INCLUDES LinkedReplyArgument (WITH COMPONENTS {
        getResult      ABSENT,
        getListError   ABSENT,
        setResult       PRESENT,
        setListError   PRESENT,
        actionResult   ABSENT,
        processingFailure PRESENT,
        deleteResult   ABSENT,
        actionError    ABSENT,
        deleteError    ABSENT } )
    ) )
m-Set OPERATION ::= localValue : 4
ROIV-m-Set ::= ROIVapdu (WITH COMPONENTS {
    invokeID      PRESENT,
    linked-ID     ABSENT,
    operation-value (m-Set),
    argument      (INCLUDES SetArgument) } )
m-Set-Confirmed OPERATION ::= localValue : 5
ROIV-m-Set-Confirmed ::= ROIVapdu (WITH COMPONENTS {
    invokeID      PRESENT,
    linked-ID     ABSENT,
    operation-value (m-Set-Confirmed),
    argument      (INCLUDES SetArgument) } )
RORS-m-Set-Confirmed ::= RORSapdu (WITH COMPONENTS {
    ... ,
    invokeID      PRESENT,
    -- result sequence -- (WITH COMPONENTS
    { operation-value (m-Set-Confirmed),
    result        (INCLUDES SetResult) } )
    -- required only if there is a single reply to the ROIV-m-Set-Confirmed ROIVapdu and data is to be returned in the
    -- RORSapdu
    } )
-- This part of the ASN.1 specification provides a definition of ROERapdu subtypes used by CMIP. The subtypes of
-- the ROERapdu define the allowed values of the error value and parameter defined by that error-value for all
-- CMIP notifications and operations.
accessDenied ERROR ::= localValue : 2
ROER-accessDenied ::= ROERapdu (WITH COMPONENTS {
    invokeID      PRESENT,
    error-value   (accessDenied) } )
-- This ROERapdu may only be returned in response to the ROIV-m-Get, ROIV-m-Set-Confirmed,
-- ROIV-m-Action-Confirmed, ROIV-m-Create and ROIV-m-Delete ROIVapdus
classInstanceConflict ERROR ::= localValue : 19
ROER-classInstanceConflict ::= ROERapdu (WITH COMPONENTS {
    invokeID      PRESENT,
    error-value   (classInstanceConflict),
    parameter     (INCLUDES BaseManagedObjectId) } )
-- This ROERapdu may only be returned in response to the ROIV-m-Get, ROIV-m-Set-Confirmed,
-- ROIV-m-Action-Confirmed, ROIV-m-Create and ROIV-m-Delete ROIVapdus

```

complexityLimitation ERROR ::= localValue : 20

ROER-complexityLimitation ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (complexityLimitation),
parameter (INCLUDES ComplexityLimitation) OPTIONAL })

-- *This ROERapdu may only be returned in response to the ROIV-m-Get, ROIV-m-Set-Confirmed,*
-- *ROIV-m-Action-Confirmed and ROIV-m-Delete ROIVapdus*

duplicateManagedObjectInstance ERROR ::= localValue : 11

ROER-duplicateManagedObjectInstance ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (duplicateManagedObjectInstance),
parameter (INCLUDES ObjectInstance) })

-- *This ROERapdu may only be returned in response to the ROIV-m-Create ROIVapdu*

getListError ERROR ::= localValue : 7

ROER-getListError ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (getListError),
parameter (INCLUDES GetListError) })

-- *This ROERapdu may only be returned in response to the ROIV-m-Get ROIVapdu*

invalidArgumentValue ERROR ::= localValue : 15

ROER-invalidArgumentValue ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (invalidArgumentValue),
parameter (INCLUDES InvalidArgumentValue) })

-- *This ROERapdu may only be returned in response to the ROIV-m-EventReport-Confirmed*
-- *and ROIV-m-Action-Confirmed ROIVapdus*

invalidAttributeValue ERROR ::= localValue : 6

ROER-invalidAttributeValue ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (invalidAttributeValue),
parameter (INCLUDES Attribute) })

-- *This ROERapdu may only be returned in response to the ROIV-m-Create ROIVapdu*

invalidFilter ERROR ::= localValue : 4

ROER-invalidFilter ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (invalidFilter),
parameter (INCLUDES CMISFilter) })

-- *This ROERapdu may only be returned in response to the ROIV-m-Get, ROIV-m-Set-Confirmed,*
-- *ROIV-m-Action-Confirmed and ROIV-m-Delete ROIVapdus*

invalidObjectInstance ERROR ::= localValue : 17

ROER-invalidObjectInstance ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (invalidObjectInstance),
parameter (INCLUDES ObjectInstance) })

-- *This ROERapdu may only be returned in response to the ROIV-m-Create ROIVapdu*

invalidScope ERROR ::= localValue : 16

ROER-invalidScope ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (invalidScope),
parameter (INCLUDES Scope) })

-- *This ROERapdu may only be returned in response to the ROIV-m-Get, ROIV-m-Set-Confirmed,*
-- *ROIV-m-Action-Confirmed and ROIV-m-Delete ROIVapdus*

missingAttributeValue ERROR ::= localValue : 18

**ROER-missingAttributeValue ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (missingAttributeValue),
 parameter (INCLUDES SET OF AttributeId) })**

-- *This ROERapdu may only be returned in response to the ROIV-m-Create ROIVapdu*

mistypedOperation ERROR ::= localValue : 21

**ROER-mistypedOperation ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (mistypedOperation) })**

-- *This ROERapdu may only be returned in response to the ROIV-m-Cancel-Get ROIVapdu*

noSuchAction ERROR ::= localValue : 9

**ROER-noSuchAction ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (noSuchAction),
 parameter (INCLUDES NoSuchAction) })**

-- *This ROERapdu may only be returned in response to the ROIV-m-Action-Confirmed ROIVapdu*

noSuchArgument ERROR ::= localValue : 14

**ROER-noSuchArgument ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (noSuchArgument),
 parameter (INCLUDES NoSuchArgument) })**

-- *This ROERapdu may only be returned in response to the ROIV-m-EventReport-Confirmed and*

-- *ROIV-m-Action-Confirmed ROIVapdus*

noSuchAttribute ERROR ::= localValue : 5

**ROER-noSuchAttribute ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (noSuchAttribute),
 parameter (INCLUDES AttributeId) })**

-- *This ROERapdu may only be returned in response to the ROIV-m-Create ROIVapdu*

noSuchEventType ERROR ::= localValue : 13

**ROER-noSuchEventType ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (noSuchEventType),
 parameter (INCLUDES NoSuchEventType) })**

-- *This ROERapdu may only be returned in response to the ROIV-m-EventReport-Confirmed ROIVapdu*

noSuchInvokeId ERROR ::= localValue : 22

**ROER-noSuchInvokeId ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (noSuchInvokeId),
 parameter (INCLUDES InvokeIDType) })**

-- *This ROERapdu may only be returned in response to the ROIV-m-Cancel-Get ROIVapdu*

noSuchObjectClass ERROR ::= localValue : 0

**ROER-noSuchObjectClass ::= ROERapdu (WITH COMPONENTS {
 invokeID PRESENT,
 error-value (noSuchObjectClass),
 parameter (INCLUDES ObjectClass) })**

-- *This ROERapdu may only be returned in response to the ROIV-m-EventReport-Confirmed, ROIV-m-Get,*

-- *ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed, ROIV-m-Create, and ROIV-m-Delete ROIVapdus*

noSuchObjectInstance ERROR ::= localValue : 1

ROER-noSuchObjectInstance ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (noSuchObjectInstance),
parameter (INCLUDES ObjectInstance) })

-- *This ROERapdu may only be returned in response to the ROIV-m-EventReport-Confirmed, ROIV-m-Get, ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed, ROIV-m-Create, and ROIV-m-Delete ROIVapdus*

noSuchReferenceObject ERROR ::= localValue : 12

ROER-noSuchReferenceObject ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (noSuchReferenceObject),
parameter (INCLUDES ObjectInstance) })

-- *This ROERapdu may only be returned in response to the ROIV-m-Create ROIVapdu*

operationCancelled ERROR ::= localValue : 23

ROER-operationCancelled ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (operationCancelled) })

-- *This ROERapdu may only be returned in response to the ROIV-m-Get ROIVapdu*

processingFailure ERROR ::= localValue : 10

ROER-processingFailure ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (processingFailure),
parameter (INCLUDES ProcessingFailure) OPTIONAL })

-- *This ROERapdu may only be returned in response to the ROIV-m-EventReport-Confirmed, ROIV-m-Get, ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed, ROIV-m-Create, and ROIV-m-Delete ROIVapdus*

setListError ERROR ::= localValue : 8

ROER-setListError ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (setListError),
parameter (INCLUDES SetListError) })

-- *This ROERapdu may only be returned in response to the ROIV-m-Set-Confirmed ROIVapdu*

syncNotSupported ERROR ::= localValue : 3

ROER-syncNotSupported ::= ROERapdu (WITH COMPONENTS {
invokeID PRESENT,
error-value (syncNotSupported),
parameter (INCLUDES CMISync) })

-- *This ROERapdu may only be returned in response to the ROIV-m-Get, ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed and ROIV-m-Delete ROIVapdus*

-- *To complete the abstract syntax specification provided in this annex, the definitions of the supporting types in 7.4 are incorporated by reference*

END -- of CMIP syntax definitions

Anexo C

Ejemplos de APDU de ROSE del CMISE

(Este anexo no es parte de esta Recomendación | Norma Internacional)

En este anexo se dan algunos ejemplos de la expansión completa de las APDU de ROSE que transportan información CMIP.

Estos ejemplos servirán de orientación a los usuarios de esta Recomendación | Norma Internacional.

-- ROIVapdu for the CMISE confirmed action operation.

```

ROIVapdu-example ::= [1] IMPLICIT SEQUENCE {
  invokeID          InvokeIDType,
  operation-value   INTEGER {m-Action-Confirmed (7)},
  argument          SEQUENCE {
    COMPONENTS OF  BaseManagedObjectId,
    accessControl  [5] AccessControl OPTIONAL,
    synchronization [6] IMPLICIT CMISync OPTIONAL,
    scope          [7] Scope DEFAULT baseObject,
    filter         CMISFilter DEFAULT and {},
  }
  actionInfo       [12] IMPLICIT SEQUENCE {
  actionType       ActionTypeId,
  actionInfoArg    [4] ANY DEFINED BY actionType OPTIONAL
  } }

```

-- RORSapdu for the CMISE confirmed action operation.

```

RORSapdu-example ::= [2] IMPLICIT SEQUENCE {
  invokeID          InvokeIDType,
  SEQUENCE {
    operation-value INTEGER {m-Action-Confirmed (7)},
    result          SEQUENCE {
      managedObjectClass ObjectClass OPTIONAL,
      managedObjectInstance ObjectInstance OPTIONAL,
      currentTime        [5] IMPLICIT GeneralizedTime OPTIONAL,
      actionReply        [6] IMPLICIT SEQUENCE {
      actionType         ActionTypeId,
      actionReplyInfo    [4] ANY DEFINED BY actionType } OPTIONAL
    }
  } }

```

-- ROIVapdu for the CMISE Linked Reply for a confirmed action operation.

```

ROIVapdu-linked-example ::= [1] IMPLICIT SEQUENCE {
  invokeID          InvokeIDType,
  linked-ID        [0] IMPLICIT InvokeIDType,
  operation-value   INTEGER {m-Action-Confirmed (7)},
  argument          CHOICE {
    actionResult     [4] IMPLICIT ActionResult,
    processingFailure [5] IMPLICIT ProcessingFailure,
    actionError      [7] IMPLICIT ActionError
  }
}

```

-- ROERapdu for the CMISE confirmed action operation when a noSuchAction error occurs.

```

ROERapdu-example ::= [3] IMPLICIT SEQUENCE {
  invokeID          InvokeIDType,
  error-value       INTEGER {noSuchAction (9)},
  parameter         SEQUENCE { managedObjectClass ObjectClass OPTIONAL,
                                actionId           ActionTypeId
  }
}

```

Anexo D**Sintaxis abstracta de guía**

(Este anexo no es parte de esta Recomendación | Norma Internacional)

La presente Recomendación | Norma Internacional importa especificaciones de la primera edición de la guía. Este anexo contiene un extracto de la Rec. X.501 del CCITT (1988) | ISO/CEI 9594-2:1990, de tal manera que en una publicación actual se mantenga una copia de la sintaxis abstracta requerida.

La norma de la guía tiene reglas de extensibilidad (especificadas en la Rec. UIT-T X.519 (1993) | ISO/CEI 9594-5:1995) que indican que si un sistema de guía de 1990 encuentra una sintaxis amplia, deberá ignorar lo que no comprenda. Esto podría tener implicaciones para las vinculaciones de nombres de objeto gestionadas OSI que tengan inscripciones en la guía a modo de progenitura de denominación, ya que los componentes de nombre distinguido relativo de la guía podrían utilizar la sintaxis ampliada. En tal caso habrá que eliminar los componentes facultativos de la sintaxis DistinguishedName amplia antes de formar el nombre distinguido de gestión OSI del objeto gestionado que tenga esa inscripción en la guía como progenitor de denominación.

Las producciones que siguen, utilizadas para la sintaxis DistinguishedName importada por el CMIP y otras normas, se reproducen en este anexo del módulo ASN.1,

InformationFramework {joint-iso-ccitt ds(5) modules(1) informationFramework(1)},

publicado originalmente en la Rec. X.501 del CCITT (1988) | ISO/CEI 9594-2:1990.

AttributeType ::= OBJECT IDENTIFIER

AttributeValue ::= ANY

AttributeValueAssertion ::= SEQUENCE {
 type AttributeType,
 assertion AttributeValue}

Name ::= CHOICE { rdnSequence RDNSequence }

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName

DistinguishedName ::= RDNSequence

RelativeDistinguishedName ::= SET OF AttributeValueAssertion

SERIES DE RECOMENDACIONES DEL UIT-T

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