



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

# UIT-T

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

# X.711

**Corrigendum 2**  
(02/2000)

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

**Corrigendum técnico 2: Revisión para incluir  
ASN.1: 1997**

Recomendación UIT-T X.711 (1997) – Corrigendum 2

(Anteriormente Recomendación del CCITT)

---

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

<b>REDES PÚBLICAS DE DATOS</b>	
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
<b>INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>	
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
<b>INTERFUNCIONAMIENTO ENTRE REDES</b>	
Generalidades	X.300–X.349
Sistemas de transmisión de datos por satélite	X.350–X.369
Redes basadas en el protocolo Internet	X.370–X.399
<b>SISTEMAS DE TRATAMIENTO DE MENSAJES</b>	X.400–X.499
<b>DIRECTORIO</b>	X.500–X.599
<b>GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS</b>	
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
<b>GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>	
Marco y arquitectura de la gestión de sistemas	X.700–X.709
<b>Servicio y protocolo de comunicación de gestión</b>	<b>X.710–X.719</b>
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
<b>SEGURIDAD</b>	X.800–X.849
<b>APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>	
Compromiso, concurrencia y recuperación	X.850–X.859
Procesamiento de transacciones	X.860–X.879
Operaciones a distancia	X.880–X.899
<b>PROCESAMIENTO DISTRIBUIDO ABIERTO</b>	X.900–X.999

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

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

**CORRIGENDUM TÉCNICO 2**

**Revisión para incluir ASN.1: 1997**

**Resumen**

En la Rec. UIT-T X.711 | ISO/IEC 9596-1 se especifican:

- los procedimientos de transmisión de información de gestión entre entidades de aplicación;
- la sintaxis abstracta del protocolo común de información de gestión (CMIP) y las reglas de codificación asociadas que han de aplicarse;
- los procedimientos para interpretar correctamente la información de control de protocolo;
- los requisitos de conformidad que debe satisfacer la aplicación de esta Recomendación | Norma Internacional.

Mediante este corrigendum técnico se revisa la Recomendación para incluir ASN.1:1997 en sus cláusulas 2.1, 2.2 y 3.3.

**Orígenes**

El corrigendum 2 a la Recomendación UIT-T X.711, preparado por la Comisión de Estudio 4 (1997-2000) del UIT-T, fue aprobado el 4 de febrero de 2000. Se publica también un texto idéntico como corrigendum técnico 2 a la 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 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 2002

Reservados todos los derechos. Ninguna parte de esta publicación puede reproducirse por ningún procedimiento sin previa autorización escrita por parte de la UIT.

## ÍNDICE

### *Página*

1)	Subcláusula 2.1 .....	1
2)	Subcláusula 2.2 .....	1
3)	Subcláusula 3.3 .....	2
4)	Subcláusula 5.2 .....	2
5)	Subcláusula 5.2.2 .....	2
6)	Subcláusula 6.2.1 .....	2
7)	Subcláusula 6.4.3 .....	2
8)	Subcláusula 6.4.4 .....	2
9)	Subcláusula 6.5.3 .....	2
10)	Subcláusula 6.5.4 .....	3
11)	Subcláusula 6.6.3 .....	3
12)	Subcláusula 6.6.4 .....	3
13)	Subcláusula 6.8.3 .....	3
14)	Subcláusula 6.8.4 .....	3
15)	Subcláusula 7.1 .....	3
16)	Subcláusula 7.2 .....	3
17)	Subcláusula 7.4 .....	3
18)	Subcláusula 7.5 .....	14
19)	Subcláusula 8.1 .....	14
20)	Anexo B .....	14
21)	Anexos C y D .....	23



## NORMA INTERNACIONAL

## RECOMENDACIÓN UIT-T

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

**CORRIGENDUM TÉCNICO 2**

**Revisión para incluir ASN.1: 1997**

**1) Subcláusula 2.1**

*Insértense las siguientes referencias en esta subcláusula:*

- Recomendación UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica.*
- Recomendación UIT-T X.681 (1997) | ISO/CEI 8824-2:1998, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de objetos de información.*
- Recomendación UIT-T X.682 (1997) | ISO/CEI 8824-3:1998, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de constricciones.*
- Recomendación UIT-T X.690 (1997) | ISO/CEI 8825-1:1998, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno: Especificación de las reglas de codificación básica, de las reglas de especificación canónica y de las reglas de codificación distinguida.*
- Recomendación UIT-T X.691 (1997) | ISO/CEI 8825-2:1998, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno: Especificación de las reglas de codificación paquetizada.*
- Recomendación UIT-T X.880 (1994) | ISO/CEI 13712-1:1995, *Tecnología de la información – Operaciones a distancia: Conceptos, modelo y notación.*
- Recomendación UIT-T X.881 (1994) | ISO/CEI 13712-2:1994, *Tecnología de la información – Operaciones a distancia: Realizaciones de interconexión de sistemas abiertos – Definición del elemento de servicio de operaciones a distancia.*
- Recomendación UIT-T X.882 (1994) | ISO/CEI 13712-3:1995, *Tecnología de la información – Operaciones a distancia: Realizaciones de interconexión de sistemas abierto: Especificación de protocolo del elemento de servicio de operaciones a distancia.*

**2) Subcláusula 2.2**

*Sustitúyase la cláusula 2.2 por lo siguiente:*

**2.2 Pares de Recomendaciones | Normas Internacionales de contenido técnico equivalente**

- Recomendación CCITT X.700 (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) Subcláusula 3.3

*Sustitúyase la cláusula 3.3 por lo siguiente:*

#### 3.3 Definiciones de operaciones a distancia

En esta Recomendación | Norma Internacional se utiliza el siguiente término definido en la Rec. UIT-T X.880 | ISO/CEI 13172-1:

- a) operación ligada.

En esta Recomendación | Norma Internacional se utilizan los siguientes términos definidos en la Rec. UIT-T X.881 | ISO/CEI 13172-2:

- a) iniciador de asociación;
- b) respondedor de asociación;
- c) elemento de servicio de operaciones a distancia;
- d) invocador;
- e) realizador.

### 4) Subcláusula 5.2

*Sustitúyase el primer párrafo por lo siguiente:*

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. UIT-T X.880 | ISO/CEI 13712-1. 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 asíncronas o síncronas según lo requiera la aplicación. La elección de la clase de operaciones es un asunto local implementado por la aplicación. Las operaciones no confirmadas del CMIP son asíncronas, resultado no comunicado. Tanto la entidad de aplicación iniciadora de la asociación como la respondedora de la asociación pueden invocar operaciones.

### 5) Subcláusula 5.2.2

*Sustituir* Rec. CCITT X.229 e ISO/CEI 9072-2 *por* Rec. UIT-T X.882 | ISO/CEI 13712-3.

### 6) Subcláusula 6.2.1

*Sustituir* Rec. CCITT X.229 e ISO/CEI 9072-2 *por* Rec. UIT-T X.882 | ISO/CEI 13712-3.

*Sustituir* ID de invocación *por* Id de invocación *en el cuadro 2*.

*Sustituir* ID de ligazón *por* Id de ligazón *en el cuadro 2*.

### 7) Subcláusula 6.4.3

*Sustituir* ID de ligazón *por* Id de ligazón *(4 veces)*.

### 8) Subcláusula 6.4.4

*Sustituir* ID de ligazón *por* Id de ligazón *(2 veces)*.

### 9) Subcláusula 6.5.3

*Sustituir* ID de ligazón *por* Id de ligazón *(4 veces)*.



**10) Subcláusula 6.5.4**

*Sustituir ID de ligazón por Id de ligazón (2 veces).*

**11) Subcláusula 6.6.3**

*Sustituir ID de ligazón por Id de ligazón (4 veces).*

**12) Subcláusula 6.6.4**

*Sustituir ID de ligazón por Id de ligazón (2 veces).*

**13) Subcláusula 6.8.3**

*Sustituir ID de ligazón por Id de ligazón (4 veces).*

**14) Subcláusula 6.8.4**

*Sustituir ID de ligazón por Id de ligazón (2 veces).*

**15) Subcláusula 7.1**

*Sustitúyase el primer párrafo por lo siguiente:*

La sintaxis abstracta se define en utilizando la notación especificada en la Rec. UIT-T X.680 | ISO/CEI 8824-1.

**16) Subcláusula 7.2**

*Sustituir ID de ligazón por Id de ligazón en el cuadro 4.*

*Suprimir la última fila del cuadro 4.*

**17) Subcláusula 7.4**

*Sustitúyase por lo siguiente:*

**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 las entidades pares del CMISE. Las PDU se especifican utilizando la ASN.1 y los objetos de información ROSE definidos en la Rec. UIT-T X.880 | ISO/CEI 13712-1.

-- *Common Management Information Protocol (CMIP)*

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

**DEFINITIONS ::= BEGIN**

-- *This ASN.1 specification has been checked for conformance with the ASN.1 standard by the OSS ASN.1 Tools*

**IMPORTS**

**ERROR, OPERATION**

**FROM Remote-Operations-Information-Objects**

{joint-iso-itu-t remote-operations(4) informationObjects(5) version1(0)}

**ROS**{, InvokeId, noInvokeId

**FROM Remote-Operations-Generic-ROS-PDUs**

{joint-iso-itu-t remote-operations(4) generic-ROS-PDUs(6) version1(0)};

```

CMIP-Operations OPERATION ::= {
    m-Action                |
    m-Action-Confirmed      |
    m-CancelGet             |
    m-Create                |
    m-Delete                |
    m-EventReport           |
    m-EventReport-Confirmed |
    m-Get                   |
    m-Linked-Reply          |
    m-Set                   |
    m-Set-Confirmed         |
}

```

```

CMIP-Confirmed-Operations OPERATION ::= {
    m-Action-Confirmed      |
    m-CancelGet            |
    m-Create               |
    m-Delete               |
    m-EventReport-Confirmed |
    m-Get                  |
    m-Set-Confirmed        |
}

```

-- CMISE error definitions

```

accessDenied ERROR ::= {
    PRIORITY    {0}
    CODE        local : 2 }

```

```

classInstanceConflict ERROR ::= {
    PARAMETER    BaseManagedObjectId
    PRIORITY     {1}
    CODE         local : 19 }

```

```

complexityLimitation ERROR ::= {
    PARAMETER    ComplexityLimitation OPTIONAL TRUE
    PRIORITY     {1}
    CODE         local : 20 }

```

```

duplicateManagedObjectInstance ERROR ::= {
    PARAMETER    ObjectInstance
    PRIORITY     {1}
    CODE         local : 11 }

```

```

getListError ERROR ::= {
    PARAMETER    GetListError
    PRIORITY     {1}
    CODE         local : 7 }

```

```

invalidArgumentValue ERROR ::= {
    PARAMETER    InvalidArgumentValue
    PRIORITY     {1}
    CODE         local : 15 }

```

```

invalidAttributeValue ERROR ::= {
    PARAMETER    Attribute
    PRIORITY     {1}
    CODE         local : 6 }

```

```

invalidFilter ERROR ::= {
    PARAMETER    CMISFilter
    PRIORITY     {1}
    CODE         local : 4 }

```

```

invalidObjectInstance ERROR ::= {
    PARAMETER    ObjectInstance
    PRIORITY     {1}
    CODE         local : 17 }

```

```

invalidScope ERROR ::= {
    PARAMETER    Scope
    PRIORITY      {1}
    CODE          local : 16 }

missingAttributeValue ERROR ::= {
    PARAMETER    SET OF AttributeId
    PRIORITY      {1}
    CODE          local : 18 }

mistypedOperation ERROR ::= {
    PRIORITY      {1}
    CODE          local : 21 }

noSuchAction ERROR ::= {
    PARAMETER    NoSuchAction
    PRIORITY      {1}
    CODE          local : 9 }

noSuchArgument ERROR ::= {
    PARAMETER    NoSuchArgument
    PRIORITY      {1}
    CODE          local : 14 }

noSuchAttribute ERROR ::= {
    PARAMETER    AttributeId
    PRIORITY      {1}
    CODE          local : 5 }

noSuchEventType ERROR ::= {
    PARAMETER    NoSuchEventType
    PRIORITY      {1}
    CODE          local : 13 }

noSuchInvokeId ERROR ::= {
    PARAMETER    InvokeIDType
    PRIORITY      {1}
    CODE          local : 22 }

noSuchObjectClass ERROR ::= {
    PARAMETER    ObjectClass
    PRIORITY      {1}
    CODE          local : 0 }

noSuchObjectInstance ERROR ::= {
    PARAMETER    ObjectInstance
    PRIORITY      {1}
    CODE          local : 1 }

noSuchReferenceObject ERROR ::= {
    PARAMETER    ObjectInstance
    PRIORITY      {1}
    CODE          local : 12 }

operationCancelled ERROR ::= {
    PRIORITY      {1}
    CODE          local : 23 }

processingFailure ERROR ::= {
    PARAMETER    ProcessingFailure OPTIONAL TRUE
    PRIORITY      {1}
    CODE          local : 10 }

setListError ERROR ::= {
    PARAMETER    SetListError
    PRIORITY      {1}
    CODE          local : 8 }

```

```

syncNotSupported ERROR ::= {
    PARAMETER    CMISSync
    PRIORITY      {1}
    CODE          local : 3 }

```

-- CMISE operations

-- Action operation (M-ACTION)

```

m-Action OPERATION ::= {
    ARGUMENT      ActionArgument
    RETURN RESULT  FALSE
    ALWAYS RESPONDS  FALSE
    CODE          local : 6 }

```

```

m-Action-Confirmed OPERATION ::= {
    ARGUMENT      ActionArgument
    RESULT         ActionResult OPTIONAL TRUE    -- this result is conditional;
                                                    -- for conditions see 8.3.3.2.9 of ITU-T Rec. X.710
    ERRORS         {accessDenied | classInstanceConflict | complexityLimitation | invalidScope |
                    invalidArgumentValue | invalidFilter | noSuchAction | noSuchArgument |
                    noSuchObjectClass | noSuchObjectInstance | processingFailure | syncNotSupported}
    LINKED         {m-Linked-Reply}
    CODE          local : 7 }

```

-- Cancel get operation (M-CANCEL-GET)

```

m-CancelGet OPERATION ::= {
    ARGUMENT      InvokeIDType
    RETURN RESULT  TRUE
    ERRORS         {mistypedOperation | noSuchInvokeId | processingFailure}
    CODE          local : 10 }

```

-- Create operation (M-CREATE)

```

m-Create OPERATION ::= {
    ARGUMENT      CreateArgument
    RESULT         CreateResult OPTIONAL TRUE    -- this result is conditional;
                                                    -- for conditions see 8.3.4.1.3 of ITU-T Rec. X.710
    ERRORS         {accessDenied | classInstanceConflict | duplicateManagedObjectInstance |
                    invalidAttributeValue | invalidObjectInstance | missingAttributeValue | noSuchAttribute |
                    noSuchObjectClass | noSuchObjectInstance | noSuchReferenceObject | processingFailure}
    CODE          local : 8 }

```

-- Delete operation (M-DELETE)

```

m-Delete OPERATION ::= {
    ARGUMENT      DeleteArgument
    RESULT         DeleteResult OPTIONAL TRUE    -- this result is conditional;
                                                    -- for conditions see 8.3.5.2.8 of ITU-T Rec. X.710
    ERRORS         {accessDenied | classInstanceConflict | complexityLimitation | invalidFilter | invalidScope |
                    noSuchObjectClass | noSuchObjectInstance | processingFailure | syncNotSupported}
    LINKED         {m-Linked-Reply}
    CODE          local : 9 }

```

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

```

m-EventReport OPERATION ::= {
    ARGUMENT      EventReportArgument
    RETURN RESULT  FALSE
    ALWAYS RESPONDS  FALSE
    CODE          local : 0 }

```

```

m-EventReport-Confirmed OPERATION ::= {
  ARGUMENT          EventReportArgument
  RESULT            EventReportResult OPTIONAL TRUE
  ERRORS            {invalidArgumentValue | noSuchArgument | noSuchEventType |
                      noSuchObjectClass | noSuchObjectInstance | processingFailure}
  CODE              local : 1 }

-- Get operation (M-GET)

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

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

m-Linked-Reply OPERATION ::= {
  ARGUMENT          LinkedReplyArgument
  CODE              local : 2 }

-- Set operations (M-SET)

m-Set OPERATION ::= {
  ARGUMENT          SetArgument
  RETURN RESULT    FALSE
  ALWAYS RESPONDS  FALSE
  CODE              local : 4 }

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

-- INFORMATION OBJECT definitions

-- While it is possible to use the Information object class definitions defined below to specify
-- Action types, Attribute types, Event Report types, and their associated ASN.1 type definitions,
-- the alternative approach using GDMO templates, as defined in ITU-T Rec. X.722 | ISO/IEC 10165-4,
-- continues to be available for use with this Recommendation | International Standard.

CMIP-ACTION ::= CLASS {
  &id              ActionTypeId UNIQUE,
  &Value            }
  WITH SYNTAX {TYPE  &Value
                ID    &id }

CMIP-ATTRIBUTE ::= CLASS {
  &id              AttributeId UNIQUE,
  &Value            }
  WITH SYNTAX {TYPE  &Value
                ID    &id }

CMIP-AVA ::= CLASS {
  &id              OBJECT IDENTIFIER UNIQUE,
  &Value            }

```

```

CMIP-EVENT ::= CLASS {
    &id          EventTypeId UNIQUE,
    &Value
    WITH SYNTAX {TYPE  &Value
                  ID    &id }

```

```

CMIP-SPECIFICERROR ::= CLASS {
    &id          OBJECT IDENTIFIER UNIQUE,
    &Value
    WITH SYNTAX {TYPE  &Value
                  ID    &id }

```

-- Supporting type definitions

```

AccessControl      ::= EXTERNAL

```

```

ActionArgument     ::= SEQUENCE {
    COMPONENTS OF BaseManagedObjectId,
    accessControl    [5] AccessControl OPTIONAL,
    synchronization  [6] IMPLICIT CMISync 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    CMIP-ACTION.&id ({ActionSet}),
        actionArgument [0] NoSuchArgument,
        argumentValue  [1] InvalidArgumentValue
    },
    ...
}

```

```

ActionInfo         ::= SEQUENCE {
    actionType    CMIP-ACTION.&id ({ActionSet}),
    actionInfoArg [4] CMIP-ACTION.&Value ({ActionSet} {@.actionType}) OPTIONAL
}

```

```

ActionReply        ::= SEQUENCE {
    actionType    CMIP-ACTION.&id ({ActionSet}),
    actionReplyInfo [4] CMIP-ACTION.&Value ({ActionSet} {@.actionType})
}

```

```

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

```

```

ActionSet CMIP-ACTION ::= {...}

```

```

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 {
    id          CMIP-ATTRIBUTE.&id      ({AttributeSet}),
    value       CMIP-ATTRIBUTE.&Value   ({AttributeSet} {@.id})
}

AttributeError ::= SEQUENCE {
    errorStatus  ENUMERATED {
        accessDenied      (2),
        noSuchAttribute    (5),
        invalidAttributeValue (6),
        invalidOperation   (24),
        invalidOperator    (25),
        ... },
    modifyOperator [2] IMPLICIT ModifyOperator OPTIONAL,
    attributeId    CMIP-ATTRIBUTE.&id ({AttributeSet}),
    attributeValue CMIP-ATTRIBUTE.&Value ({AttributeSet} {@.attributeId}) OPTIONAL
    -- value is 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,
    ...
}

AttributeSet CMIP-ATTRIBUTE ::= {...}

AttributeValueAssertion ::= SEQUENCE {
    id      CMIP-AVA.&id      ({AvaSet}),
    value   CMIP-AVA.&Value   ({AvaSet} {@.id})
}

AvaSet CMIP-AVA ::= {...}

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
}

CMISync ::= 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,
    ...
}

DistinguishedName ::= RDNSequence

EventReply ::= SEQUENCE {
    eventType              CMIP-EVENT.&id ({EventSet}),
    eventReplyInfo [8] CMIP-EVENT.&Value ({EventSet} {@.eventType}) OPTIONAL
}

EventReportArgument ::= SEQUENCE {
    managedObjectClass      ObjectClass,
    managedObjectInstance   ObjectInstance,
    eventTime               [5] IMPLICIT GeneralizedTime OPTIONAL,
    eventType               CMIP-EVENT.&id ({EventSet}),
    eventInfo               [8] CMIP-EVENT.&Value ({EventSet} {@.eventType}) OPTIONAL,
    ...
}

```



```

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

EventSet CMIP-EVENT ::= {...}

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 Attribute,
        anyString           [1] IMPLICIT Attribute,
        finalString         [2] IMPLICIT Attribute },
    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    CMIP-EVENT.&id ({EventSet}),
        eventInfo     [8] CMIP-EVENT.&Value ({EventSet} {@.eventType}) OPTIONAL
    }
}

```

InvokeIDType ::= InvokeId (ALL EXCEPT absent : NULL)

```

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         CMIP-ACTION.&id ({ActionSet}),
    ...
}

```

```

NoSuchArgument ::= CHOICE {
    actionId [0] IMPLICIT SEQUENCE {
        managedObjectClass ObjectClass OPTIONAL,
        actionType         CMIP-ACTION.&id ({ActionSet}) },
    eventId  [1] IMPLICIT SEQUENCE {
        managedObjectClass ObjectClass OPTIONAL,
        eventType          CMIP-EVENT.&id ({EventSet}) }
}

```

```

NoSuchEventType ::= SEQUENCE {
    managedObjectClass ObjectClass,
    eventType          CMIP-EVENT.&id ({EventSet}),
    ...
}

```

```

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 for 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 RDNSequence
}

```

-- 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,
    ...
}

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName

RelativeDistinguishedName ::= SET OF AttributeValueAssertion

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     CMIP-ATTRIBUTE.&id ({AttributeSet}),
        attributeValue  CMIP-ATTRIBUTE.&Value ({AttributeSet} {@.attributeId}) OPTIONAL },
        -- value is 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      CMIP-SPECIFICERROR.&id ({SpecificErrorSet}),
    errorInfo    CMIP-SPECIFICERROR.&Value ({SpecificErrorSet} {@.errorId})
}

SpecificErrorSet CMIP-SPECIFICERROR ::= {...}

-- the following type specifies the constraints to be applied when using ROSE to support CMIP

ROSEapdus ::= ROS{{InvokeIDType}, {CMIP-Operations}, {CMIP-Confirmed-Operations}}

END          -- End of CMIP syntax definitions

```

## 18) Subcláusula 7.5

*Sustituir* Remote-Operations-APDUs.ROSEapdus *por* Remote-Operations-Generic-ROS-PDUs.ROS.

*Sustituir* Rec. CCITT X.229 e ISO/CEI 9072-2 *por* Rec. UIT-T X.880 | ISO/CEI 13712-1.

*Sustituir* Rec. CCITT X.209 e ISO/CEI 8825 *por* Recomendación UIT-T X.690 | ISO/CEI 8825-1.

## 19) Subcláusula 8.1

*Sustituir* Rec. CCITT X.209 e ISO/CEI 8825 *por* Rec. UIT-T X.690 | ISO/CEI 8825-1.

*Sustituir* Rec. CCITT X.229 e ISO/CEI 9072-2 *por* Rec. UIT-T X.880 | ISO/CEI 13712-1.

*Sustituir el apartado f) por lo siguiente:*

- f) soportará la aptitud de la entidad de aplicación iniciadora de asociación y de la entidad de aplicación respondedora de asociación para invocar operaciones;

## 20) Anexo B

*Sustitúyase por lo siguiente:*

### Anexo B

#### Sintaxis ASN.1 ampliada

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

En este anexo se describe como los objetos de información OPERATION y ERROR de la Rec. UIT-T X.880 | ISO/CEI 13712-1 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**

-- This ASN.1 specification has been checked for conformance with the ASN.1 standard by the OSS ASN.1 Tools

**IMPORTS**

**ERROR, OPERATION**

**FROM Remote-Operations-Information-Objects**

{joint-iso-itu-t remote-operations(4) informationObjects(5) version1(0)}

**ROS**{, InvokeId, noInvokeId

**FROM Remote-Operations-Generic-ROS-PDUs**

{joint-iso-itu-t remote-operations(4) generic-ROS-PDUs(6) version1(0)};

**CMIP-Operations OPERATION ::= {**

<b>m-Action</b>	
<b>m-Action-Confirmed</b>	
<b>m-CancelGet</b>	
<b>m-Create</b>	
<b>m-Delete</b>	
<b>m-EventReport</b>	
<b>m-EventReport-Confirmed</b>	
<b>m-Get</b>	
<b>m-Linked-Reply</b>	
<b>m-Set</b>	
<b>m-Set-Confirmed</b>	}

```

CMIP-Confirmed-Operations OPERATION ::= {
    m-Action-Confirmed      |
    m-CancelGet             |
    m-Create                 |
    m-Delete                 |
    m-EventReport-Confirmed |
    m-Get                    |
    m-Set-Confirmed         }

```

-- INFORMATION OBJECT definitions

-- While it is possible to use the Information object class definitions defined below to specify

-- Action types, Attribute types, Event Report types, and their associated ASN.1 type definitions,

-- the alternative approach using GDMO templates, as defined in ITU-T Rec. X.722 | ISO/IEC 10165-4,

-- continues to be available for use with this Recommendation | International Standard.

```

CMIP-ACTION ::= CLASS {
    &id          ActionTypeId UNIQUE,
    &Value       }
    WITH SYNTAX {TYPE  &Value
                  ID    &id }

```

```

CMIP-ATTRIBUTE ::= CLASS {
    &id          AttributeId UNIQUE,
    &Value       }
    WITH SYNTAX {TYPE  &Value
                  ID    &id }

```

```

CMIP-AVA ::= CLASS {
    &id          OBJECT IDENTIFIER UNIQUE,
    &Value       }

```

```

CMIP-EVENT ::= CLASS {
    &id          EventTypeId UNIQUE,
    &Value       }
    WITH SYNTAX {TYPE  &Value
                  ID    &id }

```

```

CMIP-SPECIFICERROR ::= CLASS {
    &id          OBJECT IDENTIFIER UNIQUE,
    &Value       }
    WITH SYNTAX {TYPE  &Value
                  ID    &id }

```

-- the following type specifies the constraints to be applied when using ROSE to support CMIP

```

ROSEapdus ::= ROS{{InvokeIDType}, {CMIP-Operations}, {CMIP-Confirmed-Operations}}

```

-- CMISE operations

-- The following 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.&operationCode ::= local : 6

```

```

ROIV-m-Action ::= ROSEapdus (WITH COMPONENTS {
    invoke (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        linkedId      ABSENT,
        opcode        (m-Action),
        argument      (ActionArgument) } )
    })

```

```

m-Action-Confirmed OPERATION.&operationCode ::= local : 7

```

```

ROIV-m-Action-Confirmed ::= ROSEapdus (WITH COMPONENTS {
  invoke (WITH COMPONENTS {
    invokeId      (InvokeIDtype),
    linkedId      ABSENT,
    opcode        (m-Action-Confirmed),
    argument      (ActionArgument) } )
})

```

```

RORS-m-Action-Confirmed ::= ROSEapdus (WITH COMPONENTS {
  returnResult (WITH COMPONENTS {
    invokeId      (InvokeIDtype),
    result        (WITH COMPONENTS {
      opcode      (m-Action-Confirmed),
      result      (ActionResult) } ) OPTIONAL } )
  -- required only if there is a single reply to the ROIV-m-Action-Confirmed ROSEapdu
  -- and data is to be returned in the ROSEapdu
})

```

m-Cancel-Get OPERATION.&operationCode ::= local : 10

```

ROIV-m-Cancel-Get ::= ROSEapdus (WITH COMPONENTS {
  invoke (WITH COMPONENTS {
    invokeId      (InvokeIDtype),
    linkedId      ABSENT,
    opcode        (m-Cancel-Get),
    argument      (InvokeIDtype) } )
})

```

```

RORS-m-Cancel-Get ::= ROSEapdus (WITH COMPONENTS {
  returnResult (WITH COMPONENTS {
    invokeId      (InvokeIDtype) } )
  -- There is no result sequence for RORS-m-Cancel-Get
})

```

m-Create OPERATION.&operationCode ::= local : 8

```

ROIV-m-Create ::= ROSEapdus (WITH COMPONENTS {
  invoke (WITH COMPONENTS {
    invokeId      (InvokeIDtype),
    linkedId      ABSENT,
    opcode        (m-Create),
    argument      (CreateArgument) } )
})

```

```

RORS-m-Create ::= ROSEapdus (WITH COMPONENTS {
  returnResult (WITH COMPONENTS {
    invokeId      (InvokeIDtype),
    result        (WITH COMPONENTS {
      opcode      (m-Create),
      result      (CreateResult) } ) } )
})

```

m-Delete OPERATION.&operationCode ::= local : 9

```

ROIV-m-Delete ::= ROSEapdus (WITH COMPONENTS {
  invoke (WITH COMPONENTS {
    invokeId      (InvokeIDtype),
    linkedId      ABSENT,
    opcode        (m-Delete),
    argument      (DeleteArgument) } )
})

```

```

RORS-m-Delete ::= ROSEapdus (WITH COMPONENTS {
  returnResult (WITH COMPONENTS {
    invokeId      (InvokeIDtype),
    result        (WITH COMPONENTS {
      opcode      (m-Delete),

```

```

        result      (DeleteResult) } ) OPTIONAL } )
        -- required only if there is a single reply to the ROIV-m-DeleteROSEapdu
        -- and data is to be returned in the ROSEapdu
    } )

```

**m-EventReport OPERATION.&operationCode ::= local : 0**

```

ROIV-m-EventReport ::= ROSEapdus (WITH COMPONENTS {
    invoke (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        linkedId      ABSENT,
        opcode        (m-EventReport),
        argument      (EventReportArgument) } )
    } )

```

**m-EventReport-Confirmed OPERATION.&operationCode ::= local : 1**

```

ROIV-m-EventReport-Confirmed ::= ROSEapdus (WITH COMPONENTS {
    invoke (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        linkedId      ABSENT,
        opcode        (m-EventReport-Confirmed),
        argument      (EventReportArgument) } )
    } )

```

```

RORS-m-EventReport-Confirmed ::= ROSEapdus (WITH COMPONENTS {
    returnResult (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        result        (WITH COMPONENTS {
            opcode      (m-EventReport-Confirmed),
            result      (EventReportResult) } ) OPTIONAL } )
        -- required only if data is to be returned in the ROSEapdu
    } )

```

**m-Get OPERATION.&operationCode ::= local : 3**

```

ROIV-m-Get ::= ROSEapdus (WITH COMPONENTS {
    invoke (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        linkedId      ABSENT,
        opcode        (m-Get),
        argument      (GetArgument) } )
    } )

```

```

RORS-m-Get ::= ROSEapdus (WITH COMPONENTS {
    returnResult (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        result        (WITH COMPONENTS {
            opcode      (m-Get),
            result      (GetResult) } ) OPTIONAL } )
        -- required only if there is a single reply to the ROIV-m-Get ROSEapdus
    } )

```

**m-Linked-Reply OPERATION.&operationCode ::= local : 2**

```

ROIV-m-Linked-Reply ::= ROSEapdus (WITH COMPONENTS {
    invoke (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        linkedId      PRESENT,
        opcode        (m-Linked-Reply),
        argument      (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 ROSEapdus define the allowed values of the argument defined by  
-- the opcode for the specific CMIP linked reply operations.

```

ROIV-m-Linked-Reply-Action ::= ROIV-m-Linked-Reply (WITH COMPONENTS {
  invoke (WITH COMPONENTS {
    invokeId      (InvokeIDtype),
    linkedId      PRESENT,
    opcode        (m-Linked-Reply),
    argument      (LinkedReplyArgument (WITH COMPONENTS {
      invoke (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 {
  invoke (WITH COMPONENTS {
    invokeId      (InvokeIDtype),
    linkedId      PRESENT,
    opcode        (m-Linked-Reply),
    argument      (LinkedReplyArgument (WITH COMPONENTS {
      invoke (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 {
  invoke (WITH COMPONENTS {
    invokeId      (InvokeIDtype),
    linkedId      PRESENT,
    opcode        (m-Linked-Reply),
    argument      (LinkedReplyArgument (WITH COMPONENTS {
      invoke (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 {
  invoke (WITH COMPONENTS {
    invokeId      (InvokeIDtype),
    linkedId      PRESENT,
    opcode        (m-Linked-Reply),
    argument      (LinkedReplyArgument (WITH COMPONENTS {

```



```

        invoke (WITH COMPONENTS {
            getResult      ABSENT,
            getListError    ABSENT,
            setResult       PRESENT,
            setListError    PRESENT,
            actionResult    ABSENT,
            processingFailure PRESENT,
            deleteResult    ABSENT,
            actionError     ABSENT,
            deleteError     ABSENT } )
    } )
} )

```

**m-Set OPERATION.&operationcode ::= local : 4**

```

ROIV-m-Set ::= ROSEapdus (WITH COMPONENTS {
    invoke (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        linkedId      ABSENT,
        opcode        (m-Set),
        argument      (SetArgument) } )
    } )

```

**m-Set-Confirmed OPERATION.&operationCode ::= local : 5**

```

ROIV-m-Set-Confirmed ::= ROSEapdus (WITH COMPONENTS {
    invoke (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        linkedId      ABSENT,
        opcode        (m-Set-Confirmed),
        argument      (SetArgument) } )
    } )

```

```

RORS-m-Set-Confirmed ::= ROSEapdus (WITH COMPONENTS {
    returnResult (WITH COMPONENTS {
        invokeId      (InvokeIDtype),
        result        (WITH COMPONENTS {
            opcode      (m-Set-Confirmed),
            result      (SetResult) } ) OPTIONAL } )
    -- required only if there is a single reply to the ROIV-m-Set-Confirmed ROSEapdu
    -- and data is to be returned in the ROSEapdu
    } )

```

-- The following 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.&errorCode ::= local : 2**

```

ROER-accessDenied ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId      PRESENT,
        errcode       (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.&errorCode ::= local : 19**

```

ROER-classInstanceConflict ::= ROSEapdus (WITH COMPONENTS {
    returnError (WITH COMPONENTS {
        invokeId      PRESENT,
        errcode       (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.&errorCode ::= local : 20**

**ROER-complexityLimitation ::= ROSEapdus (WITH COMPONENTS {**  
**returnError (WITH COMPONENTS {**  
**invokeId     PRESENT,**  
**errcode     (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.&errorCode ::= local : 11**

**ROER-duplicateManagedObjectInstance ::= ROSEapdus (WITH COMPONENTS {**  
**returnError (WITH COMPONENTS {**  
**invokeId     PRESENT,**  
**errcode     (duplicateManagedObjectInstance),**  
**parameter   (INCLUDES ObjectInstance) } )**  
*-- This ROERapdu may only be returned in response to the ROIV-m-Create ROIVapdu*  
**} )**

**getListError ERROR.&errorCode ::= local : 7**

**ROER-getListError ::= ROSEapdus (WITH COMPONENTS {**  
**returnError (WITH COMPONENTS {**  
**invokeId     PRESENT,**  
**errcode     (getListError),**  
**parameter   (INCLUDES GetListError) } )**  
*-- This ROERapdu may only be returned in response to the ROIV-m-Get ROIVapdu*  
**} )**

**invalidArgumentValue ERROR.&errorCode ::= local : 15**

**ROER-invalidArgumentValue ::= ROSEapdus (WITH COMPONENTS {**  
**returnError (WITH COMPONENTS {**  
**invokeId     PRESENT,**  
**errcode     (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.&errorCode ::= local : 6**

**ROER-invalidAttributeValue ::= ROSEapdus (WITH COMPONENTS {**  
**returnError (WITH COMPONENTS {**  
**invokeId     PRESENT,**  
**errcode     (invalidAttributeValue),**  
**parameter   (INCLUDES Attribute) } )**  
*-- This ROERapdu may only be returned in response to the ROIV-m-Create ROIVapdu*  
**} )**

**invalidFilter ERROR.&errorCode ::= local : 4**

**ROER-invalidFilter ::= ROSEapdus (WITH COMPONENTS {**  
**returnError (WITH COMPONENTS {**  
**invokeId     PRESENT,**  
**errcode     (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.&errorCode ::= local : 17**

**ROER-invalidObjectInstance ::= ROSEapdus (WITH COMPONENTS {  
     returnError (WITH COMPONENTS {  
         invokeId     PRESENT,  
         errcode     (invalidObjectInstance),  
         parameter   (INCLUDES ObjectInstance) } )**

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

**invalidScope ERROR.&errorCode ::= local : 16**

**ROER-invalidScope ::= ROSEapdus (WITH COMPONENTS {  
     returnError (WITH COMPONENTS {  
         invokeId     PRESENT,  
         errcode     (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.&errorCode ::= local : 18**

**ROER-missingAttributeValue ::= ROSEapdus (WITH COMPONENTS {  
     returnError (WITH COMPONENTS {  
         invokeId     PRESENT,  
         errcode     (missingAttributeValue),  
         parameter   (INCLUDES SET OF AttributeId) } )**

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

**mistypedOperation ERROR.&errorCode ::= local : 21**

**ROER-mistypedOperation ::= ROSEapdus (WITH COMPONENTS {  
     returnError (WITH COMPONENTS {  
         invokeId     PRESENT,  
         errcode     (mistypedOperation) } )**

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

**noSuchAction ERROR.&errorCode ::= local : 9**

**ROER-noSuchAction ::= ROSEapdus (WITH COMPONENTS {  
     returnError (WITH COMPONENTS {  
         invokeId     PRESENT,  
         errcode     (noSuchAction),  
         parameter   (INCLUDES NoSuchAction) } )**

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

**noSuchArgument ERROR.&errorCode ::= local : 14**

**ROER-noSuchArgument ::= ROSEapdus (WITH COMPONENTS {  
     returnError (WITH COMPONENTS {  
         invokeId     PRESENT,  
         errcode     (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.&errorCode ::= local : 5**

**ROER-noSuchAttribute ::= ROSEapdus (WITH COMPONENTS {  
     returnError (WITH COMPONENTS {  
         invokeId     PRESENT,  
         errcode     (noSuchAttribute),  
         parameter   (INCLUDES AttributeId) } )**

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

**noSuchEventType ERROR.&errorCode ::= local : 13**

**ROER-noSuchEventType ::= ROSEapdus (WITH COMPONENTS {**

**returnError (WITH COMPONENTS {**  
**invokeId     PRESENT,**  
**errcode     (noSuchEventType),**  
**parameter   (INCLUDES NoSuchEventType) } )**

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

**noSuchInvokeId ERROR.&errorCode ::= local : 22**

**ROER-noSuchInvokeId ::= ROSEapdus (WITH COMPONENTS {**

**returnError (WITH COMPONENTS {**  
**invokeId     PRESENT,**  
**errcode     (noSuchInvokeId),**  
**parameter   (INCLUDES InvokeIDType) } )**

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

**noSuchObjectClass ERROR.&errorCode ::= local : 0**

**ROER-noSuchObjectClass ::= ROSEapdus (WITH COMPONENTS {**

**returnError (WITH COMPONENTS {**  
**invokeId     PRESENT,**  
**errcode     (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.&errorCode ::= local : 1**

**ROER-noSuchObjectInstance ::= ROSEapdus (WITH COMPONENTS {**

**returnError (WITH COMPONENTS {**  
**invokeId     PRESENT,**  
**errcode     (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.&errorCode ::= local : 12**

**ROER-noSuchReferenceObject ::= ROSEapdus (WITH COMPONENTS {**

**returnError (WITH COMPONENTS {**  
**invokeId     PRESENT,**  
**errcode     (noSuchReferenceObject),**  
**parameter   (INCLUDES ObjectInstance) } )**

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

**operationCancelled ERROR.&errorCode ::= local : 23**

**ROER-operationCancelled ::= ROSEapdus (WITH COMPONENTS {**

**returnError (WITH COMPONENTS {**  
**invokeId     PRESENT,**  
**errcode     (operationCancelled) } )**

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

**processingFailure ERROR.&errorCode ::= local : 10**

**ROER-processingFailure ::= ROSEapdus (WITH COMPONENTS {**

**returnError (WITH COMPONENTS {**  
**invokeId     PRESENT,**  
**errcode     (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.&errorCode ::= local : 8

ROER-setListError ::= ROSEapdus (WITH COMPONENTS {  
   returnError (WITH COMPONENTS {  
     invokeId     PRESENT,  
     errcode     (setListError),  
     parameter   (INCLUDES SetListError) } )  
 } )

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

syncNotSupported ERROR.&errorCode ::= local : 3

ROER-syncNotSupported ::= ROSEapdus (WITH COMPONENTS {  
   returnError (WITH COMPONENTS {  
     invokeId     PRESENT,  
     errcode     (syncNotSupported),  
     parameter   (INCLUDES CMISSync) } )  
 } )

-- 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

## 21) Anexos C y D

*Suprimanse los anexos C y D.*





## SERIES DE RECOMENDACIONES DEL UIT-T

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