



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

T.175

(02/98)

SERIE T: TERMINALES PARA SERVICIOS DE
TELEMÁTICA

Interfaz de programación de aplicaciones para MHEG-5

Recomendación UIT-T T.175

(Anteriormente Recomendación del CCITT)

RECOMENDACIONES DE LA SERIE T DEL UIT-T
TERMINALES PARA SERVICIOS DE TELEMÁTICA



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

RECOMENDACIÓN UIT-T T.175

INTERFAZ DE PROGRAMACIÓN DE APLICACIONES PARA MHEG-5

Resumen

Esta Recomendación especifica la interfaz de programación de aplicaciones (API) para la manipulación de objetos de información multimedios e hipermedios.

El MHEG parte 5 (Recomendación T.172) es una norma que especifica la representación codificada de objetos de información multimedios/hipermedios (objetos MHEG-5) para aplicaciones de nivel básico. Los denominados objetos MHEG-5 son tratados, interpretados y presentados por motores MHEG-5.

Orígenes

La Recomendación UIT-T T.175 ha sido preparada por la Comisión de Estudio 16 (1997-2000) del UIT-T y fue aprobada por el procedimiento de la Resolución N.º 1 de la CMNT el 6 de febrero de 1998.

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

		Página
1	Alcance	1
2	Referencias normativas.....	1
3	Definiciones y abreviaturas.....	1
3.1	Definiciones	1
3.2	Abreviaturas.....	2
4	Visión general	2
4.1	Formato de intercambio de aplicación DAVIC	2
4.2	Núcleo de las API de Java	3
5	API MHEG-5.....	4
6	Correspondencia entre las acciones elementales MHEG-5 y las operaciones API MHEG-5	22

Recomendación T.175

INTERFAZ DE PROGRAMACIÓN DE APLICACIONES PARA MHEG-5

(Ginebra, 1998)

1 Alcance

La presente Recomendación especifica la interfaz de programación de aplicaciones (API) para la manipulación de objetos de información multimedios e hipermedios, es decir la API que proveerán los motores MHEG-5 para su control por aplicaciones que funcionan en terminales conformes a DAVIC 1.1.

El MHEG parte 5 (Recomendación T.172) es una norma que especifica la representación codificada de objetos de información multimedios/hipermedios (objetos MHEG-5) para aplicaciones de nivel básico. Los denominados objetos MHEG-5 son tratados, interpretados y presentados por motores MHEG-5.

2 Referencias normativas

Las siguientes Recomendaciones del UIT-T y otras referencias contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y otras referencias son objeto de revisiones por lo que se preconiza que los usuarios de esta Recomendación investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y otras referencias citadas a continuación. Se publica periódicamente una lista de las Recomendaciones UIT-T actualmente vigentes.

- [1] ISO/IEC 13522-5:1997, *Information technology – Coding of multimedia and hypermedia information – Part 5: Support for base-level interactive applications.*
- [2] ETS 300 777-1, *Terminal Equipment (TE); End-to-end protocols for multimedia information retrieval services; Part 1: Coding of multimedia and hypermedia information for basic multimedia applications (MHEG-5).*
- [3] ETS 300 777-2, *Terminal Equipment (TE); End-to-end protocols for multimedia information retrieval services; Part 2: Use of Digital Storage Media Command and Control (DSM-CC) for basic multimedia applications.*
- [4] ISO/IEC DIS 13522-6, *Information technology – Coding of multimedia and hypermedia information – Part 6: Support for enhanced interactive applications.*

3 Definiciones y abreviaturas

3.1 Definiciones

Para los fines de la presente Recomendación se aplican las definiciones de las normas indicadas a continuación. En caso de ambigüedad, se aplican, por orden decreciente, las definiciones de las normas siguientes:

- ISO/CEI 13522-5 [1] MHEG-5;
- cualquier otra parte normativa de ISO/CEI 13522 MHEG.

3.1.1 interfaz de programación de aplicaciones (API, *application programming interface*): Frontera a través de la cual una aplicación de soporte lógico utiliza facilidades de lenguajes de programación para invocar servicios de soporte lógico. Estas facilidades pueden incluir procedimientos u operaciones, objetos de datos compartidos y resolución de identificadores.

3.1.2 aplicación local: Parte de soporte lógico que forma parte de la aplicación de (telecomunicación) y funciona en el equipo considerado.

3.1.3 interfaz de programación de aplicaciones grupo de expertos en codificación de información multimedios e hipermedios N.º 5: API proporcionada por un motor MHEG-5 a las aplicaciones locales para la manipulación de objetos MHEG-5, definidos en esta Recomendación.

3.1.4 motor grupo de expertos en codificación de información multimedios e hipermedios N.º 5: Proceso o conjunto de procesos que interpretan objetos MHEG-5 codificados de conformidad con las especificaciones de codificación ETS 300 777-1 [2] o la notación textual MHEG-5.

3.2 Abreviaturas

En esta Recomendación se utilizan las siguientes siglas.

API	Interfaz de programación de aplicaciones (<i>application programming interface</i>)
ASN.1	Notación de sintaxis abstracta uno (<i>abstract syntax notation one</i>)
DAVIC	Consejo audiovisual digital (<i>digital audio visual council</i>)
DSM-CC	Instrucción y control de medios de almacenamiento digital (<i>digital storage media command and control</i>)
EBNF	Forma Backus-Naur ampliada (<i>extended Backus-Naur form</i>)
JTC	Comité técnico mixto (<i>joint technical committee</i>)
MHEG	Grupo de expertos en codificación de información multimedios e hipermedios (<i>multimedia and hypermedia information coding experts group</i>)
SI	Información de servicio (<i>service information</i>)
STU	Unidad superior (<i>set top unit</i>)
VM	Máquina virtual (<i>virtual machine</i>)

4 Visión general

La subcláusula siguiente sitúa la API definida por esta Recomendación en el marco de las especificaciones DAVIC.

4.1 Formato de intercambio de aplicación DAVIC

Para entregar información multimedios a las STU de manera interoperable, las aplicaciones utilizarán el formato de intercambio de forma final MHEG-5, definido por ISO/CEI 13522-5 [1]. Se utilizará la codificación y la notación ASN.1 definidas por ETS 300 777-1 [2], para intercambiar objetos MHEG-5. Este formato define la semántica y la codificación de los objetos multimedios e hipermedios.

Para entregar códigos programa a las STU de manera interoperable, las aplicaciones utilizarán la clase programa intercambiado (InterchangedProgram) MHEG-5 para encapsular el código VM Java¹, de conformidad con la semántica y la codificación definidas por ISO/CEI DIS 13522-6 [4]. Las clases de VM Java son llamadas desde objetos MHEG-5, utilizando las acciones elementales llamada (*call*) y diapasón (*fork*) MHEG-5.

La unidad de intercambio de código VM Java es una clase de VM Java. Las clases de VM Java se codificarán como se define en la sección *formato de fichero de clase* de la *especificación de máquina virtual Java*. Una clase Java encapsula datos y métodos que consisten en secuencias de instrucciones. El conjunto de instrucciones viene definido por la sección *conjunto de instrucciones de máquina virtual Java* de la *especificación de máquina virtual Java*.

4.2 Núcleo de las API de Java

El código VM Java utiliza el siguiente conjunto de API en las especificaciones DAVIC 1.1 para expresar acceso a funciones básicas de la STU de manera interoperable:

- el paquete `java.lang`;
- el paquete `java.util`;
- el paquete `iso.mheg5`;
- el paquete `davic.dsmccuu`;
- el paquete `etsi.si`.

NOTA 1 – La especificación VM Java proporciona mecanismos flexibles para invocar funciones externas cuya interfaz está definida como un paquete Java. La especificación DAVIC 1.1 incluye únicamente un núcleo mínimo de paquetes necesarios para que el código VM Java sea útil en un entorno DAVIC. Se prevé que ulteriormente se normalizarán paquetes Java adicionales.

NOTA 2 – Especialmente, el paquete `java.io`, si bien no es estrictamente necesario para el funcionamiento útil del entorno VM, forma parte de las clases básicas de Java. Se prevé añadir el paquete `java.io` al núcleo DAVIC de API Java junto con una especificación adecuada de su semántica en un entorno DAVIC.

El paquete `java.lang`, definido por la *documentación API de Java*, consiste en el conjunto mínimo de clases VM Java que se requieren para utilizar el código VM Java, que admite la funcionalidad siguiente: tipos de datos básicos, objeto, operaciones matemáticas, seguridad, gestión de hilos, manipulación de cadenas, tratamiento de excepciones.

El paquete `java.util`, definido por la *documentación API de Java*, consiste en las clases VM de Java que admiten varias características de utilidades comunes a todos los programas VM Java.

El paquete `iso.mheg5`, definido por esta Recomendación, proporciona un código VM Java con acceso a objetos de interacción y presentación multimedios MHEG-5 y manipulación de los mismos, es decir acceso a los atributos dinámicos de los objetos MHEG-5 e invocación de acciones elementales sobre objetos MHEG-5.

El paquete `davic.dsmccuu`, junto con los paquetes asociados `davic.CosNaming` y `davic.CosNaming.NamingContext`, definidos por ETS 300 777-2 [3], permite que el código VM Java utilice los objetos de interfaz U-U DSM-CC para acceso a los datos de red.

El paquete `davic.dsmccuu` implementa un subconjunto de la API U-U DSM-CC. Se proporciona acceso a los siguientes servicios superiores del núcleo:

- interfaz base (*Base*): operaciones cerrar (*Close*) y destruir (*Destroy*);

¹ Java es una marca comercial o una marca registrada de Sun Microsystems, Inc.

- interfaz fichero (*File*): operaciones leer (*Read*) y escribir (*Write*);
- interfaz directorio (*Directory*): operaciones abrir (*Open*), cerrar (*Close*) y obtener (*Get*);
- interfaz cabecera de servicio (*ServiceGateway*): operaciones unir (*Attach*) y desprender (*Detach*);
- interfaz Cosnaming::NamingContext: operaciones listar (*List*) y resolver (*Resolve*);
- interfaz CosNaming::BindingIterator: operaciones siguiente (*Next_One*) y siguientes N (*Next_N*).

El paquete `etsi.si` permite que el código VM Java acceda a la información transmitida en el tren de información de servicio (SI, *service information*) DAVIC.

5 API MHEG-5

```
// -----
// Package
// -----
package iso.mheg5;
// -----
// Useful definitions
// -----
public class ObjectReference
{
    /* Attributes */
    // The groupIdIdentifier attribute is optional (it may be empty)
    public byte[] groupIdIdentifier;
    public int objectNumber;

    /* Constructors */

    public ObjectReference();
    public ObjectReference(
        int objectNumber);
    public ObjectReference(
        byte[] groupIdIdentifier,
        int objectNumber);
}

// -----
public class ContentReference
{
    /* Attributes */
    byte[] reference;

    /* Constructors */

    public ContentReference();
    public ContentReference(
        byte[] reference);
}

// -----
public class MhegException extends java.lang.Exception
{
    /* Attributes */

    // Constant declarations for exceptionCode
    public static final short TARGET_NOT_AVAILABLE = 1;
    public static final short INVALID_TARGET = 2;
    public static final short INVALID_PARAMETER = 3;

    public short exceptionCode;
```

```

        public short parameterRank;
/* Constructors */
        protected MhegException();
// Construct an MhegException with exceptionCode identified by the reason
// parameter
        public MhegException(
            short reason)
        {
            exceptionCode = reason;
            parameterRank = -1;
        }
// Construct an MhegException with exceptionCode identified by the reason
// parameter and parameterRank by the position parameter
        public MhegException(
            short reason,
            short position)
        {
            exceptionCode = reason;
            parameterRank = position;
        }
    }
// -----
abstract public class Root
{
/* Constructors */
    protected Root();
/* Methods */
// Return the reference of the Java object associated with the MHEG-5 object
// whose identification is mheg50objectReference
// If the Java object does not exist, create it first
    public static final Root getObject(
        ObjectReference mheg50objectReference)
        throws MhegException;
// Correspond to the GetAvailabilityStatus MHEG-5 elementary action
    public Boolean getAvailabilityStatus()
        throws MhegException;
// Correspond to the GetRunningStatus MHEG-5 elementary action
    public Boolean getRunningStatus()
        throws MhegException;
}
// -----
abstract public class Group extends Root
{
/* Constructors */
    protected Group();
/* Methods */
// Correspond to the SetCachePriority MHEG-5 elementary action
// The cachePriority parameter value shall be within the range [0,255]
    public void setCachePriority(
        byte cachePriority)
        throws MhegException;
// Retrieve the value of the GroupCachePriority attribute
    public Integer getCachePriority()
        throws MhegException;
}

```

```

// -----
public class Application extends Group
{
/* Constructors */
    protected Application();
/* Methods */
// Correspond to the LockScreen MHEG-5 elementary action
    public void lockScreen()
        throws MhegException;
// Correspond to the UnlockScreen MHEG-5 elementary action
    public void unlockScreen()
        throws MhegException;
// Retrieve the value of the LockCount attribute
// The wrapped returned value shall be positive or equal to zero
    public Integer getLockCount()
        throws MhegException;
// Correspond to the GetEngineSupport MHEG-5 elementary action
    public Boolean getEngineSupport(byte[] feature)
        throws MhegException;
}
// -----
public class Scene extends Group
{
/* Constructors */
    protected Scene();
/* Methods */
// Correspond to the TransitionTo MHEG-5 elementary action
    public void transitionTo()
        throws MhegException;
    public void transitionTo(
        int connectionTag)
        throws MhegException;
    public void transitionTo(
        byte transitionEffect)
        throws MhegException;
    public void transitionTo(
        int connectionTag,
        byte transitionEffect)
        throws MhegException;
// Correspond to the SetTimer MHEG-5 elementary action
    public void setTimer(
        int timerId)
        throws MhegException;
    public void setTimer(
        int timerId,
        int timerValue)
        throws MhegException;
    public void setTimer(
        int timerId,
        int timerValue,
        boolean absoluteTime)
        throws MhegException;
// Retrieve the value of the TimerPosition (in the timerPosition parameter) and
// AbsoluteTime (in the absoluteTime parameter) fields of the timer whose
// identification is timerId in the Timers attribute
// If the targetted timer exists, the method returns true otherwise false
    public boolean getTimer(
        int timerId,
        Integer timerPosition,
        Boolean absoluteTime)

```

```

        throws MhegException;
// Correspond to the SendEvent MHEG-5 elementary action
    public void sendEvent(
        ObjectReference eventSource,
        byte eventType)
        throws MhegException;
    public void sendEvent(
        ObjectReference eventSource,
        byte eventType,
        boolean eventData)
        throws MhegException;
    public void sendEvent(
        ObjectReference eventSource,
        byte eventType,
        int eventData)
        throws MhegException;
    public void sendEvent(
        ObjectReference eventSource,
        byte eventType,
        byte[] eventData)
        throws MhegException;
// Correspond to the SetCursorShape MHEG-5 elementary action
    public void setCursorShape()
        throws MhegException;
    public void setCursorShape(
        ObjectReference cursorShape)
        throws MhegException;
// Retrieve the shape of the cursor
// A returned null object reference indicates that the cursor has been removed
// from the scene
    public ObjectReference getCursorShape()
        throws MhegException;
// Correspond to the SetCursorPosition MHEG-5 elementary action
    public void setCursorPosition(
        short xCursor,
        short yCursor)
        throws MhegException;
// Correspond to the GetCursorPosition MHEG-5 elementary action
    public void getCursorPosition(
        Integer xCursor,
        Integer yCursor)
        throws MhegException;
}
// -----
abstract public class Ingredient extends Root
{

    /* Constructors */
    protected Ingredient();

    /* Methods */
// Correspond to the SetData MHEG-5 elementary action
    public void setData(
        byte[] includedContent)
        throws MhegException;
    public void setData(
        ContentReference referencedContent)
        throws MhegException;
    public void setData(
        ContentReference referencedContent,
        int contentSize)
        throws MhegException;
    public void setData(

```

```

        ContentReference referencedContent,
        byte contentCachePriority)
        throws MhegException;
    public void setData(
        ContentReference referencedContent,
        int contentSize,
        byte contentCachePriority)
        throws MhegException;

// Retrieve the value of the ContentData attribute
// The returned value shall be true and content shall be of type byte[] for an
// included content
// The returned value shall be false and content shall be of type
// ContentReference for a referenced content
// contentSize and contentCachePriority are output parameters which are valid
// only for a referenced content
    public boolean getData(
        Object content,
        Integer contentSize,
        Integer contentCachePriority)
        throws MhegException;

// Correspond to the Clone MHEG-5 elementary action
    public ObjectReference clone()
        throws MhegException;

// Correspond to the Preload MHEG-5 elementary action
    public void preload()
        throws MhegException;

// Correspond to the Unload MHEG-5 elementary action
    public void unload()
        throws MhegException;
}

// -----
public class Link extends Ingredient
{
    /* Constructors */
    protected Link();

    /* Methods */

// Correspond to the Activate MHEG-5 elementary action
    public void activate()
        throws MhegException;

// Correspond to the Deactivate MHEG-5 elementary action
    public void deactivate()
        throws MhegException;
}

// -----
public class Program extends Ingredient
{
    /* Constructors */
    protected Program();

    /* Methods */

// Correspond to the Call MHEG-5 elementary action
// Every element of the parameters array shall be of one of the following types:
// Boolean, Integer, byte[], ObjectReference or ContentReference; otherwise, an
// exception is raised
    public boolean call(
        Object[] parameters)
        throws MhegException;

// Correspond to the Fork MHEG-5 elementary action
// Every element of the parameters array shall be of one of the following types:
// Boolean, Integer, byte[], ObjectReference or ContentReference; otherwise, an

```

```

// exception is raised
    public boolean fork(
        Object[] parameters)
        throws MhegException;
// Correspond to the Stop MHEG-5 elementary action
    public void stop()
        throws MhegException;
}
// -----
public class ResidentProgram extends Program
{
    /* Constructors */
        protected ResidentProgram();
}
// -----
public class RemoteProgram extends Program
{
    /* Constructors */
        protected RemoteProgram();
}
// -----
public class InterchangedProgram extends Program
{
    /* Constructors */
        protected InterchangedProgram();
}
// -----
public class Palette extends Ingredient
{
    /* Constructors */
        protected Palette();
}
// -----
public class Font extends Ingredient
{
    /* Constructors */
        protected Font();
}
// -----
public class CursorShape extends Ingredient
{
    /* Constructors */
        protected CursorShape();
}
// -----
public class Variable extends Ingredient
{
    /* Constructors */
        protected Variable();
}
// -----

```

```

public class BooleanVariable extends Variable
{
    /* Constructors */
    protected BooleanVariable();
    /* Methods */
    // Correspond to the SetVariable MHEG-5 elementary action targetted at a
    // BooleanVariable object
    public void setVariable(
        boolean value)
        throws MhegException;
    // Retrieve the value of the variable
    public Boolean getVariable()
        throws MhegException;
}
// -----
public class IntegerVariable extends Variable
{
    /* Constructors */
    protected IntegerVariable();
    /* Methods */
    // Correspond to the SetVariable MHEG-5 elementary action targetted at an
    // IntegerVariable object
    public void setVariable(
        int value)
        throws MhegException;
    // Retrieve the value of the variable
    public Integer getVariable()
        throws MhegException;
    // Correspond to the Add MHEG-5 elementary action
    public void add(
        int value)
        throws MhegException;
    // Correspond to the Subtract MHEG-5 elementary action
    public void subtract(
        int value)
        throws MhegException;
    // Correspond to the Multiply MHEG-5 elementary action
    public void multiply(
        int value)
        throws MhegException;
    // Correspond to the Divide MHEG-5 elementary action
    public void divide(
        int value)
        throws MhegException;
    // Correspond to the Modulo MHEG-5 elementary action
    public void modulo(
        int value)
        throws MhegException;
}
// -----
public class OctetStringVariable extends Variable
{
    /* Constructors */
    protected OctetStringVariable();
    /* Methods */
    // Correspond to the SetVariable MHEG-5 elementary action targetted at an

```

```

// OctetStringVariable object
    public void setVariable(
        byte[] value)
        throws MhegException;

// Retrieve the value of the variable
    public byte[] getVariable()
        throws MhegException;

// Correspond to the Append MHEG-5 elementary action
    public void append(
        byte[] value)
        throws MhegException;
}

// -----
public class ObjectRefVariable extends Variable
{
    /* Constructors */
        protected ObjectRefVariable();

    /* Methods */
// Correspond to the SetVariable MHEG-5 elementary action targetted at an
// ObjectRefVariable object
        public void setVariable(
            ObjectReference value)
            throws MhegException;

// Retrieve the value of the variable
        public ObjectReference getVariable()
            throws MhegException;
}

// -----
public class ContentRefVariable extends Variable
{
    /* Constructors */
        protected ContentRefVariable();

    /* Methods */
// Correspond to the SetVariable MHEG-5 elementary action targetted at an
// ContentRefVariable object
        public void setVariable(
            ContentReference value)
            throws MhegException;

// Retrieve the value of the variable
        public ContentReference getVariable()
            throws MhegException;
}

// -----
abstract public class Presentable extends Ingredient
{
    /* Constructors */
        protected Presentable();

    /* Methods */
// Correspond to the Run MHEG-5 elementary action
        public void run()
            throws MhegException;

// Correspond to the Stop MHEG-5 elementary action
        public void stop()
            throws MhegException;
}

```

```

// -----
public interface TokenManager
{
/* Methods */
// Correspond to the Move MHEG-5 elementary action
    public void move(
        short movementId)
        throws MhegException;
// Correspond to the MoveTo MHEG-5 elementary action
// The index parameter value shall be within the range [0, number of elements in
// the group]
    public void moveTo(
        short index)
        throws MhegException;
// Correspond to the GetTokenPosition MHEG-5 elementary action
    public Integer getTokenPosition()
        throws MhegException;
}
// -----
public class TokenGroup extends Presentable implements TokenManager
{
/* Constructors */
    protected TokenGroup();
/* Methods */
// Correspond to the implementation of the TokenManager.move method
    public void move(
        short movementId)
        throws MhegException;
// Correspond to the implementation of the TokenManager.moveTo method
// The index parameter value shall be within the range [0, number of elements in
// the group]
    public void moveTo(
        short index)
        throws MhegException;
// Correspond to the implementation of the TokenManager.getTokenPosition method
    public Integer getTokenPosition()
        throws MhegException;
// Correspond to the CallActionSlot MHEG-5 elementary action
// The index parameter value shall be within the range [0, number of elements in
// the group]
    public void callActionSlot(
        short index)
        throws MhegException;
}
// -----
public class ListGroup extends TokenGroup
{
/* Constructors */
    protected ListGroup();
/* Methods */
// Correspond to the AddItem MHEG-5 elementary action
    public void addItem(
        short itemIndex,
        ObjectReference visibleReference)
        throws MhegException;
// Correspond to the DelItem MHEG-5 elementary action
    public void delItem(

```

```

        ObjectReference visibleReference)
        throws MhegException;
// Correspond to the GetListItem MHEG-5 elementary action
    public ObjectReference getListItem(
        short itemIndex)
        throws MhegException;
// Correspond to the GetCellItem MHEG-5 elementary action
    public ObjectReference getCellItem(
        short cellIndex)
        throws MhegException;
// Correspond to the GetItemStatus MHEG-5 elementary action
    public Boolean getItemStatus(
        short itemIndex)
        throws MhegException;
// Correspond to the SelectItem MHEG-5 elementary action
    public void selectItem(
        short itemIndex)
        throws MhegException;
// Correspond to the DeselectItem MHEG-5 elementary action
    public void deselectItem(
        short itemIndex)
        throws MhegException;
// Correspond to the ToggleItem MHEG-5 elementary action
    public void toggleItem(
        short itemIndex)
        throws MhegException;
// Correspond to the ScrollItems MHEG-5 elementary action
    public void scrollItems(
        short itemsToScroll)
        throws MhegException;
// Correspond to the SetFirstItem MHEG-5 elementary action
    public void setFirstItem(
        short itemIndex)
        throws MhegException;
// Correspond to the getFirstItem MHEG-5 elementary action
    public Integer getFirstItem()
        throws MhegException;
// Correspond to the getListSize MHEG-5 elementary action
    public Integer getListSize()
        throws MhegException;
}
// -----
abstract public class Visible extends Presentable
{
    /* Constructors */
    protected Visible();
    /* Methods */
// Correspond to the SetPosition MHEG-5 elementary action
    public void setPosition(
        short xPosition,
        short yPosition)
        throws MhegException;
// Correspond to the GetPosition MHEG-5 elementary action
    public void getPosition(
        Integer xPosition,
        Integer yPosition)
        throws MhegException;

```

```

// Correspond to the SetBoxSize MHEG-5 elementary action
// The xBoxSize and yBoxSize parameter values shall be positive and different
// from zero
    public void setBoxSize(
        short xBoxSize,
        short yBoxSize)
        throws MhegException;

// Correspond to the GetBoxSize MHEG-5 elementary action
    public void getBoxSize(
        Integer xBoxSize,
        Integer yBoxSize)
        throws MhegException;

// Correspond to the BringToFront MHEG-5 elementary action
    public void bringToFront()
        throws MhegException;

// Correspond to the SendToBack MHEG-5 elementary action
    public void sendToBack()
        throws MhegException;

// Correspond to the PutBefore MHEG-5 elementary action
    public void putBefore(
        ObjectReference visibleReference)
        throws MhegException;

// Correspond to the PutBehind MHEG-5 elementary action
    public void putBehind(
        ObjectReference visibleReference)
        throws MhegException;

// Correspond to the SetPaletteRef MHEG-5 elementary action
    public void setPaletteRef(
        ObjectReference paletteReference)
        throws MhegException;

// Retrieve the value of the PaletteRef attribute
    public ObjectReference getPaletteRef()
        throws MhegException;
}

// -----
public class Bitmap extends Visible
{
    /* Constructors */
    protected Bitmap();

    /* Methods */

// Correspond to the ScaleBitmap MHEG-5 elementary action
// The xScale and yScale parameter values shall be positive and different from
// zero
    public void scaleBitmap(
        short xScale,
        short yScale)
        throws MhegException;

// Correspond to the SetTransparency MHEG-5 elementary action
// The transparency parameter value shall be within the range [0,100]
    public void setTransparency(
        byte transparency)
        throws MhegException;

// Retrieve the value of the Transparency attribute
    public Integer getTransparency()
        throws MhegException;
}

// -----
public class LineArt extends Visible
{

```

```

/* Constructors */
    protected LineArt();
/* Methods */
// Correspond to the SetLineWidth MHEG-5 elementary action
// The lineWidth parameter value shall be positive and different from zero
    public void setLineWidth(
        byte lineWidth)
        throws MhegException;
// Correspond to the SetLineStyle MHEG-5 elementary action
// The lineStyle parameter value shall be 1 for solid, 2 for dashed, 3 for dotted
    public void setLineStyle(
        byte lineStyle)
        throws MhegException;
// Correspond to the SetLineColour MHEG-5 elementary action
    public void setLineColour(
        byte colourIndex)
        throws MhegException;
    public void setLineColour(
        byte[] absoluteColour)
        throws MhegException;
// Correspond to the SetFillColour MHEG-5 elementary action
// When neither a colour index nor an absolute colour is specified, the fill-in
// colour shall be set to transparent
    public void setFillColour()
        throws MhegException;
    public void setFillColour(
        byte colourIndex)
        throws MhegException;
    public void setFillColour(
        byte[] absoluteColour)
        throws MhegException;
}

// -----
public class Rectangle extends LineArt
{
/* Constructors */
    protected Rectangle();
}

// -----
public class DynamicLineArt extends LineArt
{
/* Constructors */
    protected DynamicLineArt();
/* Methods */
// Correspond to the GetLineWidth MHEG-5 elementary action
    public Integer getLineWidth()
        throws MhegException;
// Correspond to the GetLineStyle MHEG-5 elementary action
    public Integer getLineStyle()
        throws MhegException;
// Correspond to the GetLineColour MHEG-5 elementary action
// The returned value shall be of one of the following types: Integer (in which
// case it represents an index) or byte[] (in which case it represents the
// absolute colour)
    public Object getLineColour()
        throws MhegException;
}

```

```

// Correspond to the GetFillColour MHEG-5 elementary action
// The returned value shall be of one of the following types: Integer (in which
// case it represents an index) or byte[] (in which case it represents the
// absolute colour); a returned null object reference indicates no fill-in colour
// (transparent)
    public Object getFillColour()
        throws MhegException;

// Correspond to the DrawArc MHEG-5 elementary action
    public void drawArc(
        short x,
        short y,
        short ellipseWidth,
        short ellipseHeight,
        short startAngle,
        short arcAngle)
        throws MhegException;

// Correspond to the DrawSector MHEG-5 elementary action
    public void drawSector(
        short x,
        short y,
        short ellipseWidth,
        short ellipseHeight,
        short startAngle,
        short arcAngle)
        throws MhegException;

// Correspond to the DrawLine MHEG-5 elementary action
    public void drawLine(
        short x1,
        short y1,
        short x2,
        short y2)
        throws MhegException;

// Correspond to the DrawOval MHEG-5 elementary action
    public void drawOval(
        short x,
        short y,
        short ellipseWidth,
        short ellipseHeight)
        throws MhegException;

// Correspond to the DrawPolygon MHEG-5 elementary action
    public void drawPolygon(
        short[][] pointList)
        throws MhegException;

// Correspond to the DrawPolyline MHEG-5 elementary action
    public void drawPolyline(
        short[][] pointList)
        throws MhegException;

// Correspond to the DrawRectangle MHEG-5 elementary action
    public void drawRectangle(
        short x1,
        short y1,
        short x2,
        short y2)
        throws MhegException;

// Correspond to the Clear MHEG-5 elementary action
    public void clear()
        throws MhegException;
}

// -----
public class Text extends Visible
{

```

```

/* Constructors */
    protected Text();
/* Methods */
// Correspond to the GetTextContent MHEG-5 elementary action
// The returned value shall be of one of the following types: ContentReference
// (in which case it represents a reference to the content) or byte[] (in which
// case it represents the actual content)
    public Object getTextContent()
        throws MhegException;
// Correspond to the GetTextData MHEG-5 elementary action
    public byte[] getTextData()
        throws MhegException;
// Correspond to the SetFontRef MHEG-5 elementary action
    public void setFontRef(
        byte[] fontName)
        throws MhegException;
    public void setFontRef(
        ObjectReference fontReference)
        throws MhegException;
// Retrieve the value of the Font attribute
// The returned value shall be of one of the following types: ObjectReference (in
// which case it represents a reference to an MHEG-5 Font object) or byte[] (in
// which case it represents a resident font name); a returned null object
// reference indicates the default font
    public Object getFontRef()
        throws MhegException;
}
// -----
public class Stream extends Presentable
{
/* Constructors */
    protected Stream();
/* Methods */
// Correspond to the SetCounterTrigger MHEG-5 elementary action
// When the counter value is not specified, the targetted trigger shall be
// removed
    public void setCounterTrigger(
        short triggerIdentifier)
        throws MhegException;
    public void setCounterTrigger(
        short triggerIdentifier,
        byte counterValue)
        throws MhegException;
// Retrieve the value of the CounterPosition field of the trigger whose
// identification is triggerIdentifier in the CounterTriggers attribute; a
// returned null object reference indicates that the targetted trigger does not
// exist
    public Integer getCounterTrigger(
        short triggerIdentifier)
        throws MhegException;
// Correspond to the SetSpeed MHEG-5 elementary action
    public void setSpeed(
        byte nominator)
        throws MhegException;
    public void setSpeed(
        byte nominator,
        byte denominator)
        throws MhegException;

```

```

// Retrieve the value of the Speed attribute
    public void getSpeed(
        Integer nominator,
        Integer denominator)
        throws MhegException;

// Correspond to the SetCounterPosition MHEG-5 elementary action
    public void setCounterPosition(
        short counterPosition)
        throws MhegException;

// Retrieve the value of the CounterPosition attribute
    public Integer getCounterPosition()
        throws MhegException;

// Correspond to the SetCounterEndPosition MHEG-5 elementary action
    public void setCounterEndPosition(
        short counterEndPosition)
        throws MhegException;

// Retrieve the value of the CounterEndPosition attribute
    public Integer getCounterEndPosition()
        throws MhegException;
}

// -----
public class Audio extends Presentable
{
    /* Constructors */
        protected Audio();

    /* Methods */

// Correspond to the SetVolume MHEG-5 elementary action
    public void setVolume(
        byte volume)
        throws MhegException;

// Correspond to the GetVolume MHEG-5 elementary action
    public Integer getVolume()
        throws MhegException;
}

// -----
public class Video extends Visible
{
    /* Constructors */
        protected Video();

    /* Methods */

// Correspond to the ScaleVideo MHEG-5 elementary action
// The xScale and yScale parameter values shall be positive and different from
// zero
    public void scaleVideo(
        short xScale,
        short yScale)
        throws MhegException;
}

// -----
public class RTGraphics extends Visible
{
    /* Constructors */
        protected RTGraphics();
}

```

```

// -----
public interface Interactable
{
/* Methods */
// Correspond to the SetInteractionStatus MHEG-5 elementary action
    public void setInteractionStatus(
        boolean interactionStatus)
        throws MhegException;
// Correspond to the GetInteractionStatus MHEG-5 elementary action
    public Boolean getInteractionStatus()
        throws MhegException;
// Correspond to the SetHighlightStatus MHEG-5 elementary action
    public void setHighlightStatus(
        boolean highlightStatus)
        throws MhegException;
// Correspond to the GetHighlightStatus MHEG-5 elementary action
    public Boolean getHighlightStatus()
        throws MhegException;
}
// -----
public class slider extends Visible implements Interactable
{
/* Constructors */
    protected Slider();
/* Methods */
// Correspond to the implementation of the Interactable.setInteractionStatus
// method
    public void setInteractionStatus(
        boolean interactionStatus)
        throws MhegException;
// Correspond to the implementation of the Interactable.getInteractionStatus
// method
    public Boolean getInteractionStatus()
        throws MhegException;
// Correspond to the implementation of the Interactable.setHighlightStatus method
    public void setHighlightStatus(
        boolean highlightStatus)
        throws MhegException;
// Correspond to the implementation of the Interactable.getHighlightStatus method
    public Boolean getHighlightStatus()
        throws MhegException;
// Correspond to the Step MHEG-5 elementary action
    public void step(
        byte nbOfSteps)
        throws MhegException;
// Correspond to the SetSliderValue MHEG-5 elementary action
    public void setSliderValue(
        byte sliderValue)
        throws MhegException;
// Correspond to the GetSliderValue MHEG-5 elementary action
    public Integer getSliderValue()
        throws MhegException;
// Correspond to the SetPortion MHEG-5 elementary action
    public void setPortion(
        byte portion)
        throws MhegException;
}

```

```

// Correspond to the GetPortion MHEG-5 elementary action
    public Integer getPortion()
        throws MhegException;
}
// -----
public class EntryField extends Text implements Interactable
{
    /* Constructors */
        protected EntryField();
    /* Methods */
// Correspond to the implementation of the Interactable.setInteractionStatus
// method
        public void setInteractionStatus(
            boolean interactionStatus)
            throws MhegException;
// Correspond to the implementation of the Interactable.getInteractionStatus
// method
        public Boolean getInteractionStatus()
            throws MhegException;
// Correspond to the implementation of the Interactable.setHighlightStatus method
        public void setHighlightStatus(
            boolean highlightStatus)
            throws MhegException;
// Correspond to the implementation of the Interactable.getHighlightStatus method
        public Boolean getHighlightStatus()
            throws MhegException;
// Correspond to the SetOverwriteMode MHEG-5 elementary action
        public void setOverwriteMode(
            boolean overwriteMode)
            throws MhegException;
// Correspond to the GetOverwriteMode MHEG-5 elementary action
        public Boolean getOverwriteMode()
            throws MhegException;
// Correspond to the SetEntryPoint MHEG-5 elementary action
// The entryPoint parameter value shall be positive or equal to zero
        public void setEntryPoint(
            byte entryPoint)
            throws MhegException;
// Correspond to the GetEntryPoint MHEG-5 elementary action
        public Integer getEntryPoint()
            throws MhegException;
}
// -----
public class HyperText extends Text implements Interactable
{
    /* Constructors */
        protected HyperText();
    /* Methods */
// Correspond to the implementation of the Interactable.setInteractionStatus
method
        public void setInteractionStatus(
            boolean interactionStatus)
            throws MhegException;
// Correspond to the implementation of the Interactable.getInteractionStatus
// method
        public Boolean getInteractionStatus()
            throws MhegException;
}

```

```

// Correspond to the implementation of the Interactable.setHighlightStatus method
    public void setHighlightStatus(
        boolean highlightStatus)
        throws MhegException;

// Correspond to the implementation of the Interactable.getHighlightStatus method
    public Boolean getHighlightStatus()
        throws MhegException;

// Correspond to the GetLastAnchorFired MHEG-5 elementary action
    public byte[] getLastAnchorFired()
        throws MhegException;
}

// -----
abstract public class Button extends Visible implements Interactable
{
    /* Constructors */
        protected Button();

    /* Methods */
// Correspond to the implementation of the Interactable.setInteractionStatus
// method
        public void setInteractionStatus(
            boolean interactionStatus)
            throws MhegException;

// Correspond to the implementation of the Interactable.getInteractionStatus
// method
        public Boolean getInteractionStatus()
            throws MhegException;

// Correspond to the implementation of the Interactable.setHighlightStatus method
        public void setHighlightStatus(
            boolean highlightStatus)
            throws MhegException;

// Correspond to the implementation of the Interactable.getHighlightStatus method
        public Boolean getHighlightStatus()
            throws MhegException;

// Correspond to the Select MHEG-5 elementary action
        public void select()
            throws MhegException;

// Correspond to the Deselect MHEG-5 elementary action
        public void deselect()
            throws MhegException;
}

// -----
public class Hotspot extends Button
{
    /* Constructors */
        protected Hotspot();
}

// -----
public class PushButton extends Button
{
    /* Constructors */
        protected PushButton();

    /* Methods */
// Correspond to the SetLabel MHEG-5 elementary action
        public void setLabel(
            byte[] label)
            throws MhegException;
}

```

```
// Correspond to the GetLabel MHEG-5 elementary action
    public byte[] getLabel()
        throws MhegException;
}
// -----
public class SwitchButton extends PushButton
{

/* Constructors */
    protected SwitchButton();
/* Methods */
// Correspond to the GetSelectionStatus MHEG-5 elementary action
    public Boolean getSelectionStatus()
        throws MhegException;
// Correspond to the Toggle MHEG-5 elementary action
    public void toggle()
        throws MhegException;
}
```

6 Correspondencia entre las acciones elementales MHEG-5 y las operaciones API MHEG-5

En el cuadro 1 se describe cómo las API MHEG-5 hacen corresponder las clases MHEG-5 con clases e interfaces Java, y las acciones elementales MHEG-5 a métodos Java. El cuadro 1 no es normativo.

Cuadro 1/T.175 – Correspondencia entre acciones elementales MHEG-5 y operaciones API MHEG-5

Clase MHEG-5	Acción elemental MHEG-5	Operación API (sobrecargas)	Clase o interfaz API
Root		getObject	Root (abstract class)
	GetAvailabilityStatus	getAvailabilityStatus	
	GetRunningStatus	getRunningStatus	
Group	SetCachePriority	setCachePriority	Group (abstract class)
		getCachePriority	
Application	StorePersistent		Application (class)
	ReadPersistent		
	Launch		
	Spawn		
	Quit		
	LockScreen	lockScreen	
	UnlockScreen	unlockScreen	
		getLockCount	
	OpenConnection		
	CloseConnection		
	GetEngineSupport	getEngineSupport	

**Cuadro 1/T.175 – Correspondencia entre acciones elementales MHEG-5
y operaciones API MHEG-5 (continuación)**

Clase MHEG-5	Acción elemental MHEG-5	Operación API (sobrecargas)	Clase o interfaz API
Scene	TransitionTo	transitionTo (4)	Scene (class)
	SetTimer	setTimer (3)	
		getTimer	
	SendEvent	sendEvent (4)	
	SetCursorShape	setCursorShape (2)	
		getCursorShape	
	SetCursorPosition	setCursorPosition	
	GetCursorPosition	getCursorPosition	
Ingredient	SetData	setData (5)	Ingredient (abstract class)
		getData	
	Clone	clone	
	Preload	preload	
	Unload	unload	
Link	Activate	activate	Link (class)
	Deactivate	deactivate	
Program	Call	call	Program (abstract class)
	Fork	fork	
	Stop	stop	
ResidentProgram			ResidentProgram (class)
RemoteProgram			RemoteProgram (class)
InterchangedProgram			InterchangedProgram (class)
Palette			Palette (class)
Font			Font (class)
CursorShape			CursorShape (class)
Variable	SetVariable		Variable (abstract class)
	TestVariable		
BooleanVariable		setVariable	BooleanVariable (class)
		getVariable	

**Cuadro 1/T.175 – Correspondencia entre acciones elementales MHEG-5
y operaciones API MHEG-5 (continuación)**

Clase MHEG-5	Acción elemental MHEG-5	Operación API (sobrecargas)	Clase o interfaz API
IntegerVariable		setVariable	IntegerVariable (class)
		getVariable	
	Add	add	
	Subtract	subtract	
	Multiply	multiply	
	Divide	divide	
	Modulo	modulo	
OctetStringVariable		setVariable	OctetStringVariable (class)
		getVariable	
	Append	append	
ObjectRefVariable		setVariable	ObjectRefVariable (class)
		getVariable	
ContentRefVariable		setVariable	ContentRefVariable (class)
		getVariable	
Presentable	Run	run	Presentable (abstract class)
	Stop	stop	
TokenManager	Move	move	TokenManager (interface)
	MoveTo	moveTo	
	GetTokenPosition	getTokenPosition	
TokenGroup	CallActionSlot	callActionSlot	TokenGroup (class)
ListGroup	AddItem	addItem	TemplateGroup (class)
	DelItem	delItem	
	GetListItem	getListItem	
	GetCellItem	getCellItem	
	GetItemStatus	getItemStatus	
	SelectItem	selectItem	
	DeselectItem	deselectItem	
	ToggleItem	toggleItem	
	ScrollItems	scrollItems	
	SetFirstItem	setFirstItem	
	GetFirstItem	getFirstItem	
	GetListSize	getListSize	

**Cuadro 1/T.175 – Correspondencia entre acciones elementales MHEG-5
y operaciones API MHEG-5 (continuación)**

Clase MHEG-5	Acción elemental MHEG-5	Operación API (sobrecargas)	Clase o interfaz API
Visible	SetPosition	setPosition	Visible (abstract class)
	GetPosition	getPosition	
	SetBoxSize	setBoxSize	
	GetBoxSize	getBoxSize	
	BringToFront	bringToFront	
	SendToBack	sendToBack	
	PutBefore	putBefore	
	PutBehind	putBehind	
	SetPaletteRef	setPaletteRef	
		getPaletteRef	
Bitmap	ScaleBitmap	scaleBitmap	Bitmap (class)
	SetTransparency	setTransparency	
		getTransparency	
LineArt	SetLineWidth	setLineWidth	LineArt (class)
	SetLineStyle	setLineStyle	
	SetLineColour	setLineColour (2)	
	SetFillColour	setFillColour (3)	
Rectangle			Rectangle (class)
DynamicLineArt	GetLineWidth	getLineWidth	DynamicLineArt (class)
	GetLineStyle	getLineStyle	
	GetLineColour	getLineColour	
	GetFillColour	getFillColour	
	DrawArc	drawArc	
	DrawSector	drawSector	
	DrawLine	drawLine	
	DrawOval	drawOval	
	DrawPolygon	drawPolygon	
	DrawPolyline	drawPolyline	
	DrawRectangle	drawRectangle	
	Clear	clear	
Text	GetTextContent	getTextContent	Text (class)
	GetTextData	getTextData	
	SetFontRef	setFontRef (2)	
		getFontRef	

**Cuadro 1/T.175 – Correspondencia entre acciones elementales MHEG-5
y operaciones API MHEG-5 (fin)**

Clase MHEG-5	Acción elemental MHEG-5	Operación API (sobrecargas)	Clase o interfaz API
Stream	SetCounterTrigger	setCounterTrigger (2)	Stream (class)
		getCounterTrigger	
	SetSpeed	setSpeed (2)	
		getSpeed	
	SetCounterPosition	setCounterPosition	
		getCounterPosition	
	SetCounterEndPosition	setCounterEndPosition	
		getCounterEndPosition	
Audio	SetVolume	setVolume	Audio (class)
	GetVolume	getVolume	
Video	ScaleVideo	scaleVideo	Video (class)
RTGraphics			RTGraphics (class)
Interactable	SetInteractionStatus	setInteractionStatus	Interactable (interface)
	GetInteractionStatus	getInteractionStatus	
	SetHighlightStatus	setHighlightStatus	
	GetHighlightStatus	getHighlightStatus	
Slider	Step	step	Slider (class)
	SetSliderValue	setSliderValue	
	GetSliderValue	getSliderValue	
	SetPortion	setPortion	
	GetPortion	getPortion	
EntryField	SetOverwriteMode	setOverwriteMode	EntryField (class)
	GetOverwriteMode	getOverwriteMode	
	SetEntryPoint	setEntryPoint	
	GetEntryPoint	getEntryPoint	
HyperText	GetLastAnchorFired	getLastAnchorFired	HyperText (class)
Button	Select	select	Button (abstract class)
	Deselect	deselect	
Hotspot			Hotspot (class)
PushButton	SetLabel	setLabel	PushButton (class)
	GetLabel	getLabel	
SwitchButton	GetSelectionStatus	getSelectionStatus	SwitchButton (class)
	Toggle	toggle	

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