



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

T.175

(02/98)

SÉRIE T: TERMINAUX DES SERVICES TÉLÉMATIQUES

**Interface de programmation d'application pour
le système MHEG-5**

Recommandation UIT-T T.175

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE T
TERMINAUX DES SERVICES TÉLÉMATIQUES



Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.

RECOMMANDATION UIT-T T.175

INTERFACE DE PROGRAMMATION D'APPLICATION POUR LE SYSTEME MHEG-5

Résumé

La présente Recommandation donne la spécification de l'interface de programmation d'application (API, *application programming interface*) pour la manipulation d'objets informationnels multimédia et hypermédia.

La Recommandation T.172 (MHEG-5) est une norme qui donne la spécification de la représentation codée des objets informationnels multimédia/hypermédia échangés (objets MHEG-5) pour des applications de base. Lesdits objets MHEG-5 sont traités, interprétés et présentés par des moteurs MHEG-5.

Source

La Recommandation UIT-T T.175, élaborée par la Commission d'études 16 (1997-2000) de l'UIT-T, a été approuvée le 6 février 1998 selon la procédure définie dans la Résolution n° 1 de la CMNT.

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'UIT. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études à traiter par les Commissions d'études de l'UIT-T lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la CMNT.

Dans certains secteurs de la technologie de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

NOTE

Dans la présente Recommandation, l'expression "Administration" est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue.

DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un Membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

© UIT 1998

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT.

TABLE DES MATIÈRES

	Page
1	Domaine d'application
2	Références normatives
3	Définitions et abréviations
3.1	Définitions
3.2	Abréviations.....
4	Aperçu général
4.1	Format d'échange d'application DAVIC
4.2	Ensemble essentiel d'interfaces API Java
5	Interface API MHEG-5
6	Mappage entre les actions élémentaires MHEG-5 et les opérations API MHEG-5 ...

**INTERFACE DE PROGRAMMATION D'APPLICATION
POUR LE SYSTEME MHEG-5**

(Genève, 1998)

1 Domaine d'application

La présente Recommandation donne la spécification de l'interface de programmation d'application (API, *application programming interface*) pour la manipulation d'objets informationnels multimédia et hypermédia, c'est-à-dire l'interface API qui devra être fournie par les moteurs MHEG-5 en vue de leur contrôle par des applications tournant sur un terminal DAVIC 1.1 conforme.

La Recommandation T.172 (MHEG-5) est une norme, qui donne la spécification de la représentation codée des objets informationnels multimédia/hypermédia échangés (objets MHEG-5) pour des applications de base. Lesdits objets MHEG-5 sont traités, interprétés et présentés par des moteurs MHEG-5.

2 Références normatives

La présente Recommandation se réfère à certaines dispositions des Recommandations UIT-T et textes suivants qui de ce fait en sont partie intégrante. Les versions indiquées étaient en vigueur au moment de la publication de la présente Recommandation. Toute Recommandation ou tout texte étant sujet à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des références normatives suivantes. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée.

- [1] ISO/CEI 13522-5: 1997, *Technologies de l'information – Codage de l'information multimédia et hypermédia – Partie 5: Support pour applications interactives de niveau fondamental.*
- [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/CEI DIS 13522-6, *Technologies de l'information – Codage de l'information multimédia et hypermédia – Partie 6: Support pour les applications interactives améliorées.*

3 Définitions et abréviations

3.1 Définitions

Dans la présente Recommandation, on utilise les définitions figurant dans les normes citées ci-dessous. En cas d'ambiguïté, les définitions données dans les normes suivantes s'appliqueront par ordre décroissant:

- ISO/CEI 13522-5 [1] MHEG-5;
- toute autre partie de l'ISO/CEI 13522 MHEG.

3.1.1 interface de programmation d'application (API, *application programming interface*): frontière à travers laquelle une application logicielle utilise des éléments de langages de programmation pour invoquer des services logiciels. Ces éléments peuvent comprendre des procédures ou des opérations, des objets de données partagés et la résolution d'identificateurs.

3.1.2 application locale: sous-ensemble logiciel qui fait partie de l'application (de télécommunication) et qui tourne sur l'équipement considéré.

3.1.3 interface API MHEG-5: interface API fournie par un moteur MHEG-5 à des applications locales pour la manipulation d'objets MHEG-5, telle qu'elle est définie dans la présente Recommandation.

3.1.4 moteur MHEG-5: processus ou ensemble de processus qui interprètent les objets MHEG-5 codés conformément aux spécifications de codage de la Norme ETS 300 777-1 [2] ou à la notation textuelle MHEG-5.

3.2 Abréviations

La présente Recommandation utilise les abréviations suivantes:

API	interface de programmation d'application (<i>application programming interface</i>)
ASN.1	notation de syntaxe abstraite numéro un (<i>abstract syntax notation one</i>)
DAVIC	consortium DAVIC (<i>digital audio visual council</i>)
DSM-CC	commande et contrôle de média de stockage numérique (<i>digital storage media command and control</i>)
EBNF	formalisme de Backus-Naur étendu (<i>extended Backus-Naur form</i>)
JTC	comité technique mixte (<i>joint technical committee</i>)
MHEG	groupe d'experts pour le codage de l'information multimédia et hypermédia (<i>multimedia and hypermedia information coding experts group</i>)
SI	information de service (<i>service information</i>)
STU	dispositif d'adaptation multimédia (<i>set top unit</i>)
VM	machine virtuelle (<i>virtual machine</i>)

4 Aperçu général

Le sous-paragraphe qui suit permet de positionner l'interface API définie dans la présente Recommandation dans le cadre des spécifications DAVIC.

4.1 Format d'échange d'application DAVIC

Pour distribuer des informations multimédia à des dispositifs STU dans un environnement d'interfonctionnement, les applications doivent utiliser le format d'échange MHEG-5 de type forme finale, tel qu'il est défini dans l'ISO/CEI 13522-5 [1]. Le codage et la notation ASN.1, tels qu'ils sont définis dans la Norme ETS 300 777-1 [2], doivent être utilisés pour l'échange d'objets MHEG-5. Ce format donne la définition de la sémantique et du codage des objets multimédia et hypermédia.

Pour distribuer un code de programme à des dispositifs STU dans un environnement d'interfonctionnement, les applications doivent utiliser la classe MHEG-5 InterchangedProgram

(programme échangé) pour encapsuler le code de machine virtuelle Java¹, conformément à la sémantique et au codage définis dans la Norme ISO/CEI DIS 13522-6 [4]. Les classes de machine virtuelle Java sont appelées à partir d'objets MHEG-5 au moyen des actions élémentaires MHEG-5 Call et Fork.

L'unité d'échange de code de machine virtuelle Java est une classe de machine virtuelle Java. De telles classes doivent être codées conformément à la définition donnée dans la section relative au format de la classe File (*class file format*) de la spécification de la machine virtuelle Java (*Java virtual machine specification*). Une classe Java encapsule des données et des méthodes qui sont composées de séquences d'instructions. L'ensemble des instructions est défini dans la section relative à l'ensemble des instructions de la machine virtuelle Java (*Java virtual machine instruction set*) de la spécification de cette machine (*Java virtual machine specification*).

4.2 Ensemble essentiel d'interfaces API Java

L'ensemble d'interfaces API correspondant aux modules énumérés ci-après est utilisé par le code de machine virtuelle Java dans les spécifications DAVIC 1.1 afin d'exprimer l'accès aux fonctions de bases du dispositif STU dans un environnement d'interfonctionnement:

- le module `java.lang`;
- le module `java.util`;
- le module `iso.mheg5`;
- le module `davic.dsmccuu`;
- le module `etsi.si`.

NOTE 1 – La spécification de la machine virtuelle Java fournit des mécanismes souples permettant d'appeler des fonctions externes dont l'interface est définie comme étant un module Java. La spécification DAVIC 1.1 ne comprend qu'un ensemble essentiel minimal de modules nécessaires pour que le code de machine virtuelle Java soit utile dans un environnement DAVIC. D'autres modules Java devraient être normalisés ultérieurement.

NOTE 2 – Bien qu'à proprement parler il ne soit pas nécessaire pour la qualité de fonctionnement voulue de l'environnement de machine virtuelle, le module `java.io` fait partie des classes Java fondamentales. Il est prévu d'ajouter le module `java.io` à l'ensemble essentiel DAVIC d'interfaces API Java conjointement avec une spécification appropriée de sa sémantique dans un environnement DAVIC.

Le module `java.lang`, tel qu'il est défini dans le document relatif aux interfaces API Java (*Java API documentation*), est constitué de l'ensemble minimal de classes de machine virtuelle Java nécessaires pour faire tourner le code de machine virtuelle Java, ces classes prenant en charge les fonctionnalités suivantes: types de données de base, objet, opérations mathématiques, sécurité, gestion des chemins, manipulation des chaînes, traitement des anomalies.

Le module `java.util`, tel qu'il est défini dans le document relatif aux interfaces API Java (*Java API documentation*), est constitué de classes de machine virtuelle Java prenant en charge un certain nombre d'utilitaires communs à tous les programmes de machine virtuelle Java.

Le module `iso.mheg5`, tel qu'il est défini dans la présente Recommandation, fournit le code de machine virtuelle Java avec accès aux objets de présentation et d'interaction multimédia MHEG-5 et avec manipulation de ces objets, c'est-à-dire avec accès aux attributs dynamiques des objets MHEG-5 et avec invocation d'actions élémentaires sur les objets MHEG-5.

Le module `davic.dsmccuu`, conjointement avec les modules `davic.CosNaming` et `davic.CosNaming.NamingContext_` associés, tels qu'ils sont définis dans la Norme

¹ Java est une marque ou une marque déposée de Sun Microsystems, Inc.

ETS 300 777-2 [3], permet au code de machine virtuelle Java d'utiliser les objets interface utilisateur-utilisateur DSM-CC pour l'accès aux données de réseau.

Le module `davic.dsmccuu` met en oeuvre un sous-ensemble de l'interface API utilisateur-utilisateur DSM-CC. Il permet d'accéder aux services essentiels de dispositif STU suivants:

- interface Base: opérations Close (fermeture) et Destroy (destruction);
- interface File: opérations Read (lecture) et Write (écriture);
- interface Directory: opérations Open (ouverture), Close (fermeture) et Get (obtention);
- interface ServiceGateway: opérations Attach (rattachement) et Detach (détachement);
- interface NamingContext du module CosNaming: opérations List (listage) et Resolve (résolution);
- interface BindingIterator du module CosNaming: opérations Next_One (passage au suivant) et Next_N (passage aux N suivants).

Le module `etsi.si` permet au code de machine virtuelle Java d'accéder aux informations transmises dans le flux d'informations de service DAVIC.

5 Interface API MHEG-5

```
// -----  
// Package  
// -----  
package iso.mheg5;  
// -----  
// Useful definitions  
// -----  
public class   
{  
    /* 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   
{  
    /* 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 MhegRoot
{
    /* 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 MhegGroup 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 □□□□□□□□□□ 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 □□□□□ 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 □□□□□□□□□□ 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 □□□□ 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 □□□□□□□□ 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 [] extends Program
{
/* Constructors */
    protected ResidentProgram();
}

// -----
public class [] extends Program
{
/* Constructors */
    protected RemoteProgram();
}

// -----
public class [] extends Program
{
/* Constructors */
    protected InterchangedProgram();
}

// -----
public class [] extends Ingredient
{
/* Constructors */
    protected Palette();
}

// -----
public class [] extends Ingredient
{
/* Constructors */
    protected Font();
}

// -----
public class [] extends Ingredient
{
/* Constructors */
    protected CursorShape();
}

```

```

}
// -----
public class [] extends Ingredient
{
    /* Constructors */
    protected Variable();
}
// -----
public class [] 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 [] 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 [] 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 [] 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 [] 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 [] 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 MHEG5ElementaryAction
{
/* 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 MHEG5ElementaryAction 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 MHEG5ElementaryAction 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 □□□□□□ 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 □□□□□□ 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 □□□□□□□□ 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 □□□□□□□□□□ extends LineArt
{
    /* Constructors */
        protected Rectangle();
}
// -----
public class □□□□□□□□□□□□□□ 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 □□□□ 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 □□□□□□ 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 □□□□□ 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 □□□□□ 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 □□□□□□□□□□ extends Visible
{
    /* Constructors */
        protected RTGraphics();

```



```

}
// -----
public interface □□□□□□□□□□
{
/* 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 □□□□□□ 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 □□□□□□□□ 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 □□□□□□□□ 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 □□□□□□ 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 □□□□□□□□ extends Button
{
/* Constructors */
    protected Hotspot();
}
// -----
public class □□□□□□□□□□ 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 oooooooooooooo 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 Mappage entre les actions élémentaires MHEG-5 et les opérations API MHEG-5

Le Tableau 1 donne la description de la façon dont d'une part, les classes MHEG-5 sont mappées avec les classes et les interfaces Java et d'autre part, les actions élémentaires MHEG-5 sont mises en correspondance avec les méthodes Java, à l'interface API MHEG-5. Le Tableau 1 n'est pas normatif.

Tableau 1/T.175 – Mappage entre les actions élémentaires MHEG-5 et les opérations API MHEG-5

Classe MHEG-5	Action élémentaire MHEG-5	Opération API (surcharges)	Interface ou classe 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	

Tableau 1/T.175 – Mappage entre les actions élémentaires MHEG-5 et les opérations API MHEG-5 (suite)

Classe MHEG-5	Action élémentaire MHEG-5	Opération API (surcharges)	Interface ou classe 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	

Tableau 1/T.175 – Mappage entre les actions élémentaires MHEG-5 et les opérations API MHEG-5 (suite)

Classe MHEG-5	Action élémentaire MHEG-5	Opération API (surcharges)	Interface ou classe 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	

Tableau 1/T.175 – Mappage entre les actions élémentaires MHEG-5 et les opérations API MHEG-5 (suite)

Classe MHEG-5	Action élémentaire MHEG-5	Opération API (surcharges)	Interface ou classe 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	

Tableau 1/T.175 – Mappage entre les actions élémentaires MHEG-5 et les opérations API MHEG-5 (fin)

Classe MHEG-5	Action élémentaire MHEG-5	Opération API (surcharges)	Interface ou classe 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 DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux pour données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information
Série Z	Langages de programmation