



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

X.722

Amendement 2
(08/97)

SÉRIE X: RÉSEAUX POUR DONNÉES ET
COMMUNICATION ENTRE SYSTÈMES OUVERTS

Gestion OSI – Structure de l'information de gestion

Technologies de l'information – Interconnexion des
systèmes ouverts – Structure des informations de
gestion: directives pour la définition des objets
gérés

**Amendement 2: Ajout de l'élément syntaxique
"NO-MODIFY" et de directives complémentaires**

Recommandation UIT-T X.722 – Amendement 2

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE X
RÉSEAUX POUR DONNÉES ET COMMUNICATION ENTRE SYSTÈMES OUVERTS

RÉSEAUX PUBLICS POUR DONNÉES	
Services et fonctionnalités	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalisation et commutation	X.50–X.89
Aspects réseau	X.90–X.149
Maintenance	X.150–X.179
Dispositions administratives	X.180–X.199
INTERCONNEXION DES SYSTÈMES OUVERTS	
Modèle et notation	X.200–X.209
Définitions des services	X.210–X.219
Spécifications des protocoles en mode connexion	X.220–X.229
Spécifications des protocoles en mode sans connexion	X.230–X.239
Formulaires PICS	X.240–X.259
Identification des protocoles	X.260–X.269
Protocoles de sécurité	X.270–X.279
Objets gérés de couche	X.280–X.289
Tests de conformité	X.290–X.299
INTERFONCTIONNEMENT DES RÉSEAUX	
Généralités	X.300–X.349
Systèmes de transmission de données par satellite	X.350–X.399
SYSTÈMES DE MESSAGERIE	X.400–X.499
ANNUAIRE	X.500–X.599
RÉSEAUTAGE OSI ET ASPECTS SYSTÈMES	
Réseautage	X.600–X.629
Efficacité	X.630–X.639
Qualité de service	X.640–X.649
Dénomination, adressage et enregistrement	X.650–X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680–X.699
GESTION OSI	
Cadre général et architecture de la gestion-systèmes	X.700–X.709
Service et protocole de communication de gestion	X.710–X.719
Structure de l'information de gestion	X.720–X.729
Fonctions de gestion et fonctions ODMA	X.730–X.799
SÉCURITÉ	X.800–X.849
APPLICATIONS OSI	
Engagement, concomitance et rétablissement	X.850–X.859
Traitement transactionnel	X.860–X.879
Opérations distantes	X.880–X.899
TRAITEMENT RÉPARTI OUVERT	X.900–X.999

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

NORME INTERNATIONALE 10165-4

RECOMMANDATION UIT-T X.722

TECHNOLOGIES DE L'INFORMATION – INTERCONNEXION DES SYSTÈMES OUVERTS – STRUCTURE DES INFORMATIONS DE GESTION: DIRECTIVES POUR LA DÉFINITION DES OBJETS GÉRÉS

AMENDEMENT 2

Ajout de l'élément syntaxique "NO-MODIFY" et de directives complémentaires

Résumé

Le présent amendement à la Rec. X.722 du CCITT | ISO/CEI 10165-4 (Directives pour la définition des objets gérés) traite de deux sujets.

L'un des sujets de cet amendement est la spécification d'un élément syntaxique "NO-MODIFY" pour les modèles d'ensemble GDMO.

L'autre sujet de cet amendement représente des directives complémentaires qui décrivent les règles permettant de faire migrer de la notation ASN.1:1990 à la notation ASN.1:1994 des modules de l'information de gestion fondée sur la série X.700. Ces directives complémentaires offrent un moyen généralisé et cohérent pour faciliter cette migration, de sorte qu'une méthode unique est fournie aux concepteurs pour spécifier l'utilisation de l'ASN.1:1994 dans les Recommandations qu'ils mettent en œuvre.

Source

La Recommandation X.722, Amendement 2, de l'UIT-T a été approuvée le 9 août 1997. Un texte identique est publié comme Norme internationale ISO/CEI 10165-4.

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 des technologies 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

	<i>Page</i>
1) Paragraphe 2.1.....	1
2) Paragraphe 8.2.....	1
3) Paragraphe 8.4.2.....	1
4) Paragraphe 8.4.3.2.....	2
5) Index	2
6) Nouvel article 9.....	2
7) Nouvel article 10.....	8

Introduction

Le présent amendement ajoute l'élément syntaxique "NO-MODIFY" aux propriétés d'attribut du modèle d'ensemble pour spécifier clairement qu'un attribut ne peut pas être modifié dans les sous-classes et dans les classes compatibles de la classe des objets gérés.

Le présent amendement propose de **ne pas interdire l'utilisation de la notation ASN.1:1994** (c'est-à-dire de permettre l'emploi de la spécification normative de l'ASN.1) dans les normes de gestion des systèmes OSI, nouvelles et en cours de réalisation en raison:

- des erreurs et défauts de l'ASN.1:1990;
- des améliorations que contient l'ASN.1:1994.

Les modules en ASN.1:1990 et en ASN.1:1994 peuvent être utilisés ensemble; ils sont entièrement compatibles si l'on suit les directives au A.2 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, ASN.1 – Spécification de la notation de base. Ce mode d'utilisation mixte impose certaines restrictions aux productions permises, qu'elles soient en ASN.1:1990 ou en ASN.1:1994. Si les présentes directives sont respectées, les versions de 1990 et de 1994 sont non seulement entièrement compatibles, mais pratiquement indiscernables l'une de l'autre, avec des codages identiques. Par ces directives, les normes de gestion des systèmes OSI, nouvelles et en cours de réalisation, peuvent utiliser indifféremment les deux versions de l'ASN.1 pour la spécification normative de la syntaxe.

Le présent amendement introduit des conventions pour la spécification de l'ASN.1 et des directives GDMO pour identifier clairement les spécifications et les choix d'utilisation liés aux modules ASN.1 et aux modèles GDMO.

NORME INTERNATIONALE

RECOMMANDATION UIT-T

**TECHNOLOGIES DE L'INFORMATION – INTERCONNEXION DES SYSTÈMES
OUVERTS – STRUCTURE DES INFORMATIONS DE GESTION: DIRECTIVES
POUR LA DÉFINITION DES OBJETS GÉRÉS**

AMENDEMENT 2

Ajout de l'élément syntaxique "NO-MODIFY" et de directives complémentaires

1) Paragraphe 2.1

Ajouter les références suivantes par ordre numérique:

- Recommandation UIT-T X.680 (1994) | ISO/CEI 8824-1:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base.*
- Recommandation UIT-T X.681 (1994) | ISO/CEI 8824-2:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des objets informationnels.*
- Recommandation UIT-T X.682 (1994) | ISO/CEI 8824-3:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des contraintes.*
- Recommandation UIT-T X.683 (1994) | ISO/CEI 8824-4:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: paramétrage des spécifications de la notation de syntaxe abstraite numéro un.*
- Recommandation UIT-T X.690 (1994) | ISO/CEI 8825-1:1995, *Technologies de l'information – Règles de codage de la notation de syntaxe abstraite numéro un: spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives.*
- Recommandation UIT-T X.691 (1995) | ISO/CEI 8825-2:1995, *Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage compact.*

2) Paragraphe 8.2

Ajouter ce qui suit après le point m):

(Les directives pour la production des modules ASN.1:1990 et ASN.1:1994 équivalents figurent à l'article 9.)

3) Paragraphe 8.4.2

Sous *propertylist*, ajouter après [SET-BY-CREATE] (*L'élément [SET-BY-CREATE] a été ajouté dans l'Amendement 1*):

[NO-MODIFY]

4) **Paragraphe 8.4.3.2**

Ajouter les deux nouveaux alinéas suivants avant le dernier:

L'absence de la propriété REPLACE peut être utilisée pour spécifier qu'un attribut ne peut pas être remplacé pour des instances d'une classe, mais cette absence n'exclut pas des sous-classes ajoutant la propriété REPLACE. La propriété NO-MODIFY est présente pour spécifier explicitement qu'un attribut ne peut pas être modifié (qu'il peut uniquement être lu) dans la classe comportant cette propriété et dans toutes les sous-classes et tous les objets gérés compatibles (c'est-à-dire les objets gérés se comportant comme des allomorphes envers cette classe). Cette propriété est incompatible avec une définition de classe d'objet géré (et ne doit pas y figurer) ayant une des propriétés REPLACE, GET-REPLACE, ADD, REMOVE ou ADD-REMOVE pour le même attribut.

NOTE 3 – La propriété NO-MODIFY n'est pas nécessairement incompatible avec la propriété REPLACE-WITH-DEFAULT car cette opération est souvent utilisée avec un sens de "réinitialisation" qui peut être compatible avec l'incapacité, pour un gestionnaire, de contrôler la valeur de l'attribut.

NOTE 4 – Avant que la propriété NO-MODIFY ait été ajoutée aux directives GDMO, l'usage consistait à spécifier cette propriété dans les modèles BEHAVIOUR ou dans les documents cités en référence dans ceux-ci.

Si l'on souhaite qu'une partie de la définition d'un attribut ait pour effet que l'attribut ne soit pas remplacé dans toute classe qui spécifie cet attribut, il faut spécifier cette contrainte dans un modèle BEHAVIOUR cité en référence dans le modèle ATTRIBUTE.

5) **Index**

Ajouter la nouvelle rubrique suivante dans l'index:

NO-MODIFY 26-27

Tenir compte des modifications des pages concernant les rubriques suivantes de l'index:

BEHAVIOUR 25-26, 30-32, 34-35, 37-40

REPLACE 14, 26-27

6) **Nouvel article 9**

Ajouter un nouvel article 9 après le paragraphe 8.11.3.5:

9 **Directives pour créer des modules équivalents en ASN.1:1994 et en ASN.1:1990**

Dans l'élaboration de normes, on peut avoir affaire à des modules normatifs en ASN.1:1994; mais pour pouvoir les utiliser, il est recommandé de disposer d'un module normatif équivalent en ASN.1:1990 à condition:

- 1) qu'il ait le même identificateur d'objet que le module ASN.1:1994;
- 2) qu'il soit normatif et que la norme spécifie qu'en cas d'incohérence entre un module ASN.1:1990 et un module ASN.1:1994, c'est ce dernier qui prévaut;
- 3) qu'il soit précisé dans la norme que l'utilisation de l'ASN.1:1990 sera poursuivie aussi longtemps qu'il le faudra.

NOTE 1 – Les règles de l'ISO/CEI JTC 1/SC 21 imposent une révision annuelle¹⁾ portant sur le renouvellement des normes internationales en ASN.1:1990. Les organismes nationaux sont priés d'en tenir compte lorsqu'ils réviseront les normes en ASN.1:1990. On veillera ainsi à conserver les normes ASN.1:1990 aussi longtemps qu'il le faudra.

¹⁾ L'ISO/CEI JTC 1/SC 21 a confirmé que les normes en ASN.1:1990 resteraient disponibles pour des raisons de conformité et d'interprétabilité (dans le Document 9001 Rév). Le SC 21 a demandé à ses groupes de travail de poursuivre la maintenance de ces normes. Une résolution du SC 21 visant à poursuivre la maintenance sera établie à chaque réunion du SC 21 (actuellement une fois par an).

De plus, pour limiter les erreurs, il est recommandé que chaque module en ASN.1:1990 soit une transformation faite par machine d'un module en ASN.1:1994, cette opération pouvant être facilement automatisée.

NOTE 2 – Si un rédacteur souhaite utiliser, pour faire une conversion d'ASN.1:1994 en ASN.1:1990 en limitant les risques d'erreur, un outil commercial (mis à la disposition de la communauté de normalisation pour ainsi dire gratuitement, les outils ASN.1 du fournisseur XXX par exemple), il conviendrait qu'il fasse précéder le code ainsi produit d'une indication du genre:

```
-- conversion d'ASN.1:1994 en ASN.1:1990 faite au moyen --
-- d'outils ASN.1 du fournisseur XXX --
```

accompagnée de la Note suivante:

NOTE – L'ISO ne peut déclarer qu'un outil logiciel soit préférable à un autre, mais pour l'heure l'outil ASN.1 de XXX fait partie des logiciels permettant de faire la conversion d'ASN.1:1994 en ASN.1:1990.

On peut éviter des difficultés si l'on utilise uniquement le sous-ensemble commun aux deux versions de l'ASN.1. Dans ce cas, seul le module ASN.1:1994 doit être inclus dans la présente norme.

9.1 Directives

On suivra les directives ci-après:

- 1) un document de gestion de systèmes peut se référer à des modules en notation ASN.1:1990 comme en notation ASN.1:1994; mais chaque module individuel doit être entièrement conforme *soit* à l'une, *soit* à l'autre. Les directives de l'article 10 sont utilisées pour identifier la version de la notation qui est utilisée dans un module donné;
- 2) des références à des types et à des valeurs peuvent être importées dans un module ASN.1:1994 à partir d'un module ASN.1:1990 à condition:
 - a) de ne pas importer de macros ASN.1:1990 dans un module ASN.1:1994; pour cette raison, on ne peut pas créer d'instance de **MACRO** dans un module ASN.1:1994;
 - b) que les identificateurs des valeurs de **SET**, **SEQUENCE** et **CHOICE** soient présents;
- 3) par ailleurs, des références à des types ou à des valeurs peuvent être importées dans un module ASN.1:1990 à partir d'un module ASN.1:1994 à condition:
 - de ne pas importer de types **CHARACTER STRING**, **BMPStrings**, **UniversalStrings**, **EMBEDDED PDV** de l'ASN.1:1994. Du fait qu'il n'existe pas d'équivalent ASN.1:1990 de ces types ASN.1:1994, leur emploi est déconseillé dans les modules ASN.1:1994 qui requièrent un module ASN.1:1990 équivalent. Pour le même motif, l'utilisation du type ASN.1:1994 **Tuple** est interdite dans les modules ASN.1:1994 qui requièrent un module ASN.1:1990 équivalent²⁾;

NOTE 1 – Si les directives proposées sont tenues à jour, on peut importer sans inconvénient des références à des types ou à des valeurs d'une version ASN.1 dans une autre version car il y a toujours une structure équivalente.

- 4) on utilisera le module ASN.1:1994 suivant pour définir des classes d'objets informationnels ASN.1 qui sont importées dans les modules ASN.1:1994:

```
-- <ASN1.Version 1994 SModule {joint-iso-itu-t ms(9) smi(1) part4(4)
-- asn1Module(2) 2} >--
```

```
SModule {joint-iso-itu-t ms(9) smi(1) part4(4) asn1Module(2) 2}
```

```
DEFINITIONS ::= BEGIN
```

```
REGISTERED-AS ::= TYPE-IDENTIFIER
```

```
-- TYPE-IDENTIFIER est défini dans l'ISO/CEI 8824-1 et il est disponible dans chaque module sans
-- qu'il soit nécessaire de l'importer; il est défini en tant que :
```

```
-- TYPE-IDENTIFIER ::= CLASS
-- {
-- &id OBJECT IDENTIFIER UNIQUE,
-- &Type
-- }
```

²⁾ **Tuple** est le nom de la production ASN.1:1994 qui permet l'insertion de caractères de commande dans la notation de la valeur d'un IA5String, ce qui ne peut se faire en ASN.1:1990. Par exemple, ... **greetings IA5String ::= {"hello", cr, "there"}** insère un retour de chariot entre "hello" et "there" (cr est importé d'un module défini dans la Rec. UIT-T X.680 | ISO/CEI 8824-1 et est assimilé à un retour de chariot littéral).

```
-- WITH SYNTAX {&Type IDENTIFIED BY &id}
INFO-REPLY-IDENTIFIER ::= CLASS
    {
        &Info OPTIONAL,
        &Reply OPTIONAL,
        &registeredAs OBJECT IDENTIFIER UNIQUE
    }
WITH SYNTAX {INFO &Info REPLY &Reply IDENTIFIED BY &registeredAs}
RegisteredAsTable REGISTERED-AS ::= {...}
InfoReplyTable INFO-REPLY-IDENTIFIER ::= {...}
-- RegisteredAsTable doit être rempli par les modèles GDMO ATTRIBUTE et PARAMETER
-- Templates
-- InfoReplyTable doit être rempli par les modèles GDMO ACTION et NOTIFICATION Templates
END
```

- 5) les éditeurs définissent les modules en ASN.1:1994 comme dans l'exemple suivant:

```
-- <ASN1.Version 1994 ExampleModule > --
ExampleModule {-- insérer ici un identificateur d'objet valide --}
DEFINITIONS ::= BEGIN
IMPORTS
REGISTERED-AS,
INFO-REPLY-IDENTIFIER,
RegisteredAsTable,
InfoReplyTable
FROM SMMModule {joint-iso-itu-t ms(9) smi(1) part4(4) asn1Module(2) 2};
Foo ::= SEQUENCE {
    id1 REGISTERED-AS.&id ({RegisteredAsTable}),
    syntax1 REGISTERED-AS.&Type ({RegisteredAsTable} {@id1})
Bar ::= SEQUENCE {
    id2 REGISTERED-AS.&id ({RegisteredAsTable}),
    syntax2 SEQUENCE OF REGISTERED-AS.&Type ({RegisteredAsTable} {@id2})
firstExtensionId OBJECT IDENTIFIER ::= {1 3 17 103 10 1}
FirstExtensionInfo ::= PrintableString
-- Illustre l'utilisation d'une contrainte de sous-type contenu de type ouvert --
FooBar ::= Foo (WITH COMPONENTS {
    id1 (firstExtensionId),
    syntax1(FirstExtensionInfo)} )
END
```

Etant donné que les directives GDMO sont utilisées en combinaison avec la classe des objets informationnels **REGISTERED-AS** de l'ASN.1, **FooBar** fait double emploi. Autrement dit, la spécification GDMO plus **REGISTERED-AS** en ASN.1 est équivalente à la contrainte de type interne sur **Foo**. **FooBar** illustre simplement que si un type ouvert peut être contraint à n'importe quel type, un élément **ANY/ANY DEFINED BY** ne peut être contraint qu'à un type **ANY/ANY DEFINED BY** en ASN.1:1990. **FooBar** illustre également la manière de mettre en correspondance un type ouvert ainsi contraint en ASN.1:1994 avec un commentaire ASN.1 en ASN.1:1990;

- 6) la conversion d'un module ASN.1:1994 en module ASN.1:1990 équivalent doit être faite en tenant compte des instructions suivantes:
- retirer la partie de la déclaration **IMPORTS** qui se réfère au module SMMModule. Cela élimine les définitions importées des classes d'objets informationnels **REGISTERED-AS** et **INFO-REPLY-IDENTIFIER**;
 - convertir toutes les références de type ouvert en types **ANY** ou **ANY DEFINED BY**. A cet effet, convertir toute la syntaxe ASN.1 de cette forme conformément à ce qui suit:

En *premier* lieu, convertir:

DE	EN
REGISTERED-AS.&id	OBJECT IDENTIFIER

Si "REGISTERED-AS.&Type" est une composante d'un élément SET ou SEQUENCE et s'il est défini dans le même élément SET ou SEQUENCE que "id", alors convertir:

DE	EN
REGISTERED-AS.&Type ({RegisteredAsTable} {@id1})	ANY DEFINED BY id
REGISTERED-AS.&Type ({RegisteredAsTable} {@.id1})	ANY DEFINED BY id

sinon convertir:

DE	EN
REGISTERED-AS.&Type ({RegisteredAsTable} {@id1})	ANY
REGISTERED-AS.&Type ({RegisteredAsTable} {@.id1})	ANY

- c) si l'élément **AUTOMATIC TAGS** est en vigueur pour le module en ASN.1:1994, appliquer les 22.5 à 22.7 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, qui décrit la manière dont l'étiquetage automatique est appliqué aux composantes d'un type SET, SEQUENCE ou CHOICE, puis retirer l'élément syntaxique "AUTOMATIC TAGS" de la déclaration de définition du module;
- d) si un type ouvert est contraint par l'utilisation de la notation du sous-type TypeConstraint, supprimer cette contrainte étant donné que ANY et ANY DEFINED BY ne peuvent être limités que par ANY ou ANY DEFINED BY en ASN.1:1990;
- e) si la définition d'un type **ENUMERATED** utilise l'élément syntaxique "identifiant" pour son item d'énumération, le remplacer par "identifiant(number)". Par exemple, remplacer:

ENUMERATED {a, b, c, d}

par

ENUMERATED {a(0), b(1), c(2), d(3)}

- f) supprimer toutes les occurrences de marqueurs d'extension (telles que "..."). Par exemple, remplacer:

```
SEQUENCE {
    i    IA5String,
    b    BOOLEAN,
    ...
}
```

par

```
SEQUENCE {
    i    IA5String,
    b    BOOLEAN
}
```

- g) supprimer toute occurrence de l'élément **EXTENSIBILITY IMPLIED** de la déclaration de définition du module;

- h) supprimer tous les blancs et tous les caractères de retour à la ligne des chaînes hstring et bstring et supprimer tous les caractères de retour à la ligne des chaînes cstring. Par exemple, remplacer:

```

b BIT STRING ::= '0001 1100 1101 0110 1110 0111 1101 0101'B
o OCTET STRING ::= '8F3CE483 0192B345 932D5EF2 8AA3E700'H
p PrintableString ::= "Hello,
                        world!"
    
```

par

```

b BIT STRING ::= '00011100110101101110011111010101'B
o OCTET STRING ::= '8F3CE4830192B345932D5EF28AA3E700'H
p PrintableString ::= "Hello,world!"
    
```

- i) convertir toutes les notations CharacterStringList en leur équivalent cstring. Par exemple, remplacer:

```

name PrintableString ::= {"This is a long string, that is
                            spread across two lines"}
    
```

par

```

name PrintableString ::= "This is a long string, that is spread across two lines"
    
```

- j) convertir toutes les références à des jeux de valeur en références à des types contraints. Par exemple, remplacer:

```

Ages INTEGER ::= {1 | 4 | 7..20}
    
```

par

```

Ages ::= INTEGER (1 | 4 | 7..20)
    
```

NOTE 2 – Il est préférable d'utiliser des types contraints plutôt que des ensembles de valeur, à moins que le paramétrage et les classes d'objets informationnels ne soient largement utilisés, ce qui est généralement le cas quand les ensembles de valeurs sont particulièrement utiles.

- k) convertir toutes les occurrences du type **INSTANCE OF** en leur type équivalent **SEQUENCE**. Par exemple, remplacer:

```

A ::= INSTANCE OF REGISTERED-AS
    
```

par

```

A ::= SEQUENCE {
    type-id          OBJECT IDENTIFIER,
    value            [0] ANY DEFINED BY type-id
}
    
```

- l) supprimer les identificateurs "**mantissa**", "**base**" et "**exponent**" de toute valeur de type **REAL**, et remplacer toute limite de type interne dans les types **REAL** par un commentaire. Par exemple, remplacer:

```

ten REAL ::= {mantissa 1, base 10, exponent 1}
    
```

```

DecimalReal ::= REAL (WITH COMPONENTS {..., base 10})
    
```

par

```

ten REAL ::= {1, 10, 1}
    
```

```

DecimalReal ::= REAL -- codé en base 10
    
```

- m) remplacer toutes les occurrences de la notation de valeur pour **EXTERNAL** par leurs équivalents ASN.1:1990. Par exemple, remplacer:

```

extern1990 EXTERNAL ::= {
    direct-reference   { 1 2 3 4 5 6 },
    indirect-reference 3,
    encoding           single-ASN1-type : IA5String : "hello"
}
    
```

par

```
extern1994 EXTERNAL ::= {
  identification context-negotiation : {
    presentation-context-id      3,
    transfer-syntax               { 1 2 3 4 5 6 }
  },
  data-value                     notation : IA5String : "hello"
}
```

- n) remplacer tous les alphabets permis par l'ASN.1:1994 par leurs équivalents ASN.1:1990. Par exemple, remplacer:

```
UpperCaseAndSpaceOnly ::= PrintableString (FROM("A".."Z" | " "))
```

par

```
UpperCaseAndSpaceOnly ::= PrintableString (FROM("A" | "B" | "C" |
  "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" |
  "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" |
  "T" | "U" | "V" | "W" | "X" | "Y" | "Z" ))
```

- o) remplacer toute expression d'ensemble en notation ASN.1:1994, utilisée dans la notation de sous-type, par son équivalent ASN.1:1990. Par exemple, remplacer:

```
PartNumber ::= NumericString (SIZE(8) ^ FROM("0".."9"))
```

par

```
PartNumber ::= NumericString (SIZE(8)) (FROM("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"))
```

- p) quand l'alinéa o) n'est pas applicable parce qu'il se traduit par un ensemble infini, remplacer la partie de la notation de sous-type ASN.1:1994 qui en résulte par un ensemble infini accompagné d'un commentaire. Par exemple, remplacer:

```
AllButZeroToTen ::= INTEGER(ALL EXCEPT (0..10))
```

par

```
AllButZeroToTen ::= INTEGER -- tous les entiers sauf de 0 à 10
```

Le fait d'appliquer les instructions de conversion ci-dessus à **ExampleModule** en ASN.1:1994 de la directive 5 produit:

```
-- <ASN1.Version 1990 ExampleModule > --
```

```
ExampleModule {-- insérer ici une valeur valide d'identificateur --}
```

```
DEFINITIONS ::= BEGIN
```

```
Foo ::= SEQUENCE {
  id1 OBJECT IDENTIFIER,
  syntax1 ANY DEFINED BY id1}

```

```
Bar ::= SEQUENCE {
  id2 OBJECT IDENTIFIER,
  syntax2 ANY }

```

```
-- On notera que, dans ExampleModule de l'ASN.1:1994, syntax2 dans Bar est défini comme une
-- SEQUENCE OF, non comme un type ouvert, et ne peut donc être
-- converti en un ANY DEFINED BY.
```

```
firstExtensionId OBJECT IDENTIFIER ::= {1 3 17 103 10 1}
```

```
FirstExtensionInfo ::= PrintableString
```

```
-- On notera que FirstExtensionId et FirstExtensionInfo doivent être utilisés
-- avec le type Foo, où firstExtensionId est la valeur de "id1"
-- (OBJECT IDENTIFIER) qui indique que syntax1 a
-- le type de syntaxe de FirstExtensionInfo (PrintableString).
```

```
END
```

7) **Nouvel article 10**

Ajouter le nouvel article 10 suivant, après l'article 9:

10 **Conventions relatives à l'ASN.1 et aux directives GDMO**

Le présent article est consacré aux conventions afin d'identifier clairement les spécifications et les options d'utilisateur associées aux modèles GDMO et modules ASN.1 connexes. Cela s'effectue par l'emploi de directives en ligne. Ces conventions peuvent également être utiles en tant que directives pour le compilateur de notation ASN.1 et de directives GDMO. Ces conventions, si elles sont utilisées, ne requièrent pas de spécificateur pour modifier les spécifications ASN.1 et GDMO en vue de l'utilisation de ces directives. Autrement dit, les directives peuvent se trouver dans les mêmes corps de texte que les modules ASN.1 ou les modèles GDMO ou dans toute autre spécification associée. Par ailleurs, si l'on utilise ces conventions, elles ne modifient pas la spécification de l'ASN.1 et des directives GDMO. Autrement dit, ces directives ne changent pas la syntaxe de la spécification de l'ASN.1 et des directives GDMO. Ces conventions ne sont pas obligatoires, mais leur emploi est recommandé.

Ces directives sont structurées de manière à répondre à divers besoins:

- les fichiers d'entrée comportant des directives (c'est-à-dire des modules ASN.1, des bibliothèques GDMO et/ou des fichiers directives distincts) doivent être acceptables par les compilateurs ASN.1 et par les outils/compilateurs GDMO qui ne reconnaissent pas ces directives;
- les fichiers d'entrée ne comportant pas de directives doivent être acceptables par les compilateurs ASN.1 et par les outils/compilateurs GDMO.

NOTE – Ces conventions permettent l'apport de compléments, au moyen de directives spécifiques à une réalisation.

Chaque spécification de directive est un commentaire structuré contenant une directive unique formée d'un mot clé qualifié par un domaine et suivie ou non d'opérandes. La hauteur de chaque caractère (majuscule ou minuscule) a une signification. En conséquence, une directive est traitée comme un commentaire par tout outil ou compilateur qui ne prend pas en charge ces directives.

On tient compte des conventions suivantes pour décrire les directives:

- le texte en gras (par exemple `--<` ou **ASN.1**) doit être inséré exactement comme indiqué, sans aucun espace blanc supplémentaire ou changement du type de caractère (majuscule ou minuscule) par rapport à celui qui est indiqué;
- le texte en italique (par exemple *keyword*) doit être remplacé par le texte approprié;
- les éléments des directives en option doivent être mis entre crochets (par exemple [**operands**]);
- un élément de directive pouvant être répété (peu importe le nombre de fois) est suivi de trois points (`. . .`), (par exemple [**operands**] `. . .`).

Le format général d'une directive est:

```
--< directive >--
```

où la directive est:

```
Scope.keyword [operand] [ , operand ] . . .
```

Autrement dit, une directive est introduite par deux traits d'union consécutifs et un symbole «plus petit que» (`--<`), et elle est terminée par le symbole «plus grand que» suivi de deux traits d'union consécutifs (`>--`). Aucun espace ne doit être laissé entre les traits d'union et le symbole plus petit que ou plus grand que. Tout compilateur qui ne prend pas en charge ces directives les traite par conséquent comme des commentaires ASN.1 ou GDMO. Une directive ne peut contenir aucun commentaire ASN.1.

Chaque structure `--< >--` est un commentaire structuré contenant une directive unique qui est formée d'un mot clé, qualifié par un Scope, suivie ou non d'opérandes. Le type du caractère (majuscule ou minuscule) est significatif.

Les directives peuvent occuper plusieurs lignes, dont les deux premiers caractères autres que des blancs sont deux traits d'union consécutifs (`--`).

Les opérandes sont délimités par un espace blanc (un ou plusieurs espaces ou tabulateurs consécutifs) pour des éléments dissemblables, et par une virgule dans le cas d'une liste d'éléments similaires (tels que des noms de modules actifs). On peut avoir un espace blanc (zéro ou plusieurs espaces ou tabulations) avant ou après d'autres éléments d'une directive. Dans ce contexte, les caractères de retour chariot, de retour à la ligne ou de tabulation verticale ne sont PAS considérés comme des espaces blancs et ne sont pas autorisés.

10.1 Conventions relatives aux directives ASN.1

Les conventions additionnelles suivantes s'appliquent à une directive ASN.1.

Une directive ne doit contenir aucun commentaire ASN.1. Une directive peut se trouver dans le même fichier que l'item ASN.1. Lorsqu'elle est dans le même fichier, la directive peut être hors du domaine du module ASN.1 (avant le début ou après la fin). Une directive peut être placée dans le corps d'un module, si l'espace blanc le permet. Si tel est le cas, elle doit précéder l'item ASN.1 auquel elle s'applique.

Le symbole Scope d'une directive ASN.1 est **ASN1**. D'autres symboles de type Scope (attribués en fonction de la réalisation, par exemple) peuvent être définis.

Le mot clé peut être l'un des suivants:

- **Version.**

Généralement, un opérande peut être:

- une chaîne de texte (sans espace blanc, virgule, -- ou >);
- une chaîne de chiffres (sans espace blanc, virgule, -- ou >);
- une chaîne de caractères cstring ("xxxxx") dont tous les caractères sont sur la même ligne;
- { *ObjectIdentifieur* };
- éventuellement toute autre structure ASN.1, telle qu'une chaîne binaire ou hexadécimale.

10.1.1 Directive Version

La directive **version** sert à indiquer qu'un module ASN.1 est écrit conformément à la norme ASN.1:1990 ou ASN.1:1994.

Cette directive a le format suivant:

```
--<ASN1.Version version moduleName [ oid ] >--
```

Les éléments en gras (tels que **ASN1.version**) peuvent être écrits tels quels et les éléments en italique (tels que *version*) doivent être remplacés comme suit:

- *version* – **1990** ou **1994** ou **1990, 1994**.
- *moduleName* – Nom d'un module ASN.1.
- *oid* – Une valeur d'identificateur d'objet ASN.1 en option, utilisée pour identifier de manière univoque un module ASN.1.

On ne donnera qu'une seule directive **version** par module ASN.1. La directive **version 1990, 1994** signifie que le module ASN.1 se déclare conforme à l'ASN:1990 et à l'ASN.1:1994; un compilateur peut utiliser sa connaissance de l'une ou l'autre version, ou le sous-ensemble commun aux deux versions. En l'absence de cette directive, une réalisation pourra utiliser un autre moyen pour identifier la version du module, par exemple une option de compilateur dans la ligne de commande ou la reconnaissance d'une syntaxe particulière à une version donnée (par exemple une macro ASN.1:1990 ou un objet informationnel ASN.1:1994).

Exemples:

```
--<ASN1.Version 1990 Attribute-ASN1Module
-- {joint-iso-itu-t ms(9) smi(3) part2(2) asn1Module(2) 1} >--
```

```
--<ASN1.Version 1994 SMModule
-- {joint-iso-itu-t ms(9) smi(3) part4(4) asn1Module(2) 2} >--
```

10.2 Conventions sur les directives GDMO

Les conventions additionnelles suivantes s'appliquent aux directives GDMO.

Une directive ne peut contenir aucun commentaire en ASN.1.

Une directive peut se trouver dans un fichier autre que celui qui contient le modèle GDMO auquel elle se réfère, ou dans le même fichier que le modèle GDMO. Dans ce dernier cas, la directive peut être hors du domaine du document GDMO (avant le début d'un document ou après sa fin). Une directive peut se trouver dans le corps d'un document quand un espace blanc le permet.

Quand elle est dans un document, une directive doit être placée avant le modèle GDMO auquel elle s'applique. Pour un modèle GDMO donné, on peut avoir plusieurs directives différentes, mais une seule de chaque. On peut se référer, par

exemple, au même modèle par les directives `Nickname` et `Working Set`, mais une de chaque seulement. L'on peut avoir des directives `Nickname` pour d'autres items ASN.1.

Le symbole `Scope` (domaine) est une directive `GDMO`. De plus, on peut définir d'autres symboles de domaine (attribués au niveau de la réalisation, par exemple).

Le mot clé peut être l'un des suivants:

- `Alias`;
- `Document`;
- `EndDocument`;
- `Version`.

En général, un opérande peut être:

- une chaîne de mots (sans espace, virgule, `--` ou `>`);
- une chaîne de nombres (sans espace, virgule, `--` ou `>`);
- une chaîne de caractères ("`xxxxx`") sur une même ligne;
- `{ ObjectIdentifieur }`;
- éventuellement d'autres structures ASN.1 telles qu'une chaîne binaire ou hexadécimale.

10.2.1 Directive `Alias`

La directive `Alias` sert à fournir des identificateurs de remplacement, ou pseudonymes, pour un document `GDMO`. Ces pseudonymes sont utilisés pour aligner les références entre documents `GDMO` quand on a utilisé des identificateurs courts ou non systématiques.

Le format de la directive est:

```
--<GDMO.Alias documentIdentifieur documentAlias
-- [ , documentAlias ] ... >--
```

Les éléments en gras (tels que `GDMO.Alias`) doivent être écrits tels quels et les éléments en italique (tels que *documentIdentifieur*) doivent être remplacés comme indiqué ci-après.

On peut fournir un nombre quelconque d'items `documentAlias`, tous séparés par des virgules.

- *documentIdentifieur* – Un identificateur de document `GDMO`, sous la forme d'une chaîne entre guillemets (") ou d'un identificateur d'objet entre accolades ({ }).
- *documentAlias* – Un identificateur de remplacement pour un document `GDMO`, sous la forme d'une chaîne entre guillemets (") ou d'un identificateur d'objet entre accolades ({ }).

Exemples:

```
--<GDMO.Alias  "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992"
-- "Rec. X.721 | ISO/IEC 10165-2:1992",
-- "CCITT Rec. X.721 | ISO/IEC 10165-2 ",
-- "Rec. X.721| ISO/IEC 10165-2",
-- "CCITT Rec. X.721 (1992)",
-- "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992",
-- "DMI">--

--<GDMO.Alias  "Recommendation M.3100:1992"
-- "Rec. M.3100:1992",
-- "M.3100:1992",
-- "M.3100">--
```

10.2.2 Directive Document

La directive **Document** fournit l'identificateur d'un "document" GDMO où l'identificateur est une chaîne de caractères, un identificateur d'objet ou les deux. Cette directive peut avoir trois formats différents:

```
--<GDMO.Document documentString >--
--<GDMO.Document documentOID >--
--<GDMO.Document documentString documentOID >--
```

Les éléments en gras (tels que **GDMO.Document**) doivent être écrits tels quels et les éléments en italique (tels que *documentString*) doivent être remplacés par ce qui suit:

- *documentString* – La chaîne de caractères qui identifie un document GDMO, entre guillemets (").
- *documentOID* – Un identificateur d'objet ASN.1 attribué à un document GDMO.

Une directive **Document** doit se situer avant le premier modèle GDMO ou module ASN.1 constituant le document. Autrement dit, on considère qu'un document GDMO est composé de tous les modèles GDMO et modules ASN.1 suivant une directive **Document** jusqu'à la directive **EndDocument** correspondante, jusqu'à la directive **Document** suivante ou jusqu'à la fin du fichier contenant le texte GDMO.

Les règles suivantes d'enregistrement de texte GDMO dans des fichiers comportent les directives correspondantes:

- chaque document GDMO doit avoir une directive **Document** pour fournir le nom "correct" du document. En l'absence de cette directive, le nom du document n'est pas défini ou est fourni par un autre mécanisme;
- une directive **Document** doit se trouver dans le même fichier que le texte GDMO auquel elle s'applique.

Le même nom de document GDMO peut apparaître dans plusieurs directives **Document** et peut s'appliquer à plusieurs unités de texte GDMO, telles que des fichiers distincts. Dans ce cas, on considère que l'ensemble du texte GDMO se trouvant dans des fichiers différents fait partie du même document GDMO. Cela est analogue à l'espace nominatif en C++ qui permet à plusieurs fichiers d'en-tête distincts de contenir les éléments définis dans le même espace nominatif.

Généralement, il faut un seul document GDMO par fichier d'entrée. Le seul cas dans lequel un fichier peut contenir plusieurs documents GDMO est celui dans lequel le fichier a les directives **Document** et **EndDocument** appropriées.

Un espace blanc (un ou plusieurs espaces ou caractères de tabulation consécutifs) n'est pas jugé significatif dans un élément *documentIdentifier* ou *documentAlias*.

Exemples:

```
--<GDMO.Document "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992">--
--<GDMO.Document "Recommendation M.3100:1992">--
--<GDMO.Document "OP1 Library Vol. 4">--
--<GDMO.Document {iso(1) 2 124 360501 15 13 1} >--
--<GDMO.Document "IIMC MIB Translation" {iso(1) 2 124 360501 15 13 1} >--
```

10.2.3 Directive End Document

La directive **EndDocument** sert à indiquer la fin d'un document GDMO. Elle peut avoir quatre formats différents:

```
--<GDMO.EndDocument >--
--<GDMO.EndDocument documentString >--
--<GDMO.EndDocument documentOID >--
--<GDMO.EndDocument documentString documentOID >--
```

Les éléments en gras (tels que **GDMO.EndDocument**) doivent être écrits tels quels et les éléments en italique (tels que *documentString*) doivent être remplacés comme suit:

- *documentString* – La chaîne de caractères, entre guillemets ("), qui identifie un document GDMO.
- *documentOID* – Un identificateur d'objet ASN.1, entre accolades ({ }), attribué à un document GDMO.

Un document GDMO est composé de tous les modèles GDMO et modules ASN.1 suivant une directive **Document** jusqu'à la directive **EndDocument** correspondante, jusqu'à la directive **Document** suivante, ou jusqu'à la fin du fichier contenant le texte GDMO. Autrement dit, l'utilisation de la directive **EndDocument** n'est pas requise. Si une directive **EndDocument** est utilisée, elle doit figurer après le dernier des modèles GDMO ou des modules ASN.1 qui composent le document. Les éléments *documentString* et/ou *documentOID* d'une directive **EndDocument** doivent correspondre à ceux de la directive **Document** précédente.

Exemples:

```
--<GDMO.EndDocument >--
--<GDMO.EndDocument "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992">--
--<GDMO.EndDocument "Recommendation M.3100:1992">--
--<GDMO.EndDocument "OP1 Library Vol. 4">--
--<GDMO.EndDocument {iso(1) 2 124 360501 15 13 1} >--
--<GDMO.EndDocument "IIMC MIB Translation" {iso(1) 2 124 360501 15 13 1} >--
```

10.2.4 Directive Version

La directive **Version** sert à indiquer la version des directives GDMO à laquelle correspond un document. Elle peut avoir l'un des deux formats suivants:

```
--<GDMO.Version version >--
--<GDMO.Version version documentIdentifieur >--
```

Les éléments en gras (tels que **GDMO.Version**) doivent être écrits tels quels et les éléments en italique (tels que *version*) doivent être remplacés comme suit:

- *version* - Le nombre qui indique la version GDMO selon le tableau ci-après:

Indice	Définition de la version
1	GDMO, 1992
1.1	Amendement 1, SET-BY-CREATE
1.2	Amendement 2, NO-MODIFY , ASN.1:1994 et directives
1.3	Amendement 3, Utilisation de Z dans le comportement de la classe d'objets gérés

Une nouvelle version est définie à chaque mise à jour GDMO (par exemple, révision, corrigendum technique, amendement). La prise en charge d'une version donnée est cumulative. Autrement dit, tout document GDMO valable pour la version *n* sera également valable pour toutes les versions antérieures. Par exemple, **GDMO,Version 1.2** indique que le document GDMO prend en charge GDMO 1992 (version **1**) et l'amendement **SET-BY-CREATE** (version **1.1**), outre l'amendement relatif à **NO-MODIFY**, à l'ASN.1:1994 et aux directives (version **1.2**).

- *documentIdentifieur* - L'identificateur d'un document GDMO, sous la forme d'une chaîne entre guillemets (") ou d'un identificateur d'objet entre accolades ({ }).

La première forme (sans identificateur de document) doit se situer dans le document GDMO mais avant tout modèle GDMO.

Exemples:

```
--<GDMO.Version 1 "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2:1992" >--
--<GDMO.Version 1.1 "Recommendation M.3100" >--
--<GDMO.Version 1.2 >--
```

SÉRIES 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