



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Z.100

Corrigendum 1
(08/2004)

SÉRIE Z: LANGAGES ET ASPECTS GÉNÉRAUX
LOGICIELS DES SYSTÈMES DE
TÉLÉCOMMUNICATION

Techniques de description formelle – Langage de
description et de spécification (SDL)

SDL: langage de description et de spécification

Corrigendum 1

Recommandation UIT-T Z.100 (2002) – Corrigendum 1

RECOMMANDATIONS UIT-T DE LA SÉRIE Z
LANGAGES ET ASPECTS GÉNÉRAUX LOGICIELS DES SYSTÈMES DE TÉLÉCOMMUNICATION

TECHNIQUES DE DESCRIPTION FORMELLE	
Langage de description et de spécification (SDL)	Z.100–Z.109
Application des techniques de description formelle	Z.110–Z.119
Diagrammes des séquences de messages	Z.120–Z.129
Langage étendu de définition d'objets	Z.130–Z.139
Notation de test et de commande de test	Z.140–Z.149
Notation de prescriptions d'utilisateur	Z.150–Z.159
LANGAGES DE PROGRAMMATION	
CHILL: le langage de haut niveau de l'UIT-T	Z.200–Z.209
LANGAGE HOMME-MACHINE	
Principes généraux	Z.300–Z.309
Syntaxe de base et procédures de dialogue	Z.310–Z.319
LHM étendu pour terminaux à écrans de visualisation	Z.320–Z.329
Spécification de l'interface homme-machine	Z.330–Z.349
Interfaces homme-machine orientées données	Z.350–Z.359
Interfaces homme-machine pour la gestion des réseaux de télécommunication	Z.360–Z.369
QUALITÉ	
Qualité des logiciels de télécommunication	Z.400–Z.409
Aspects qualité des Recommandations relatives aux protocoles	Z.450–Z.459
MÉTHODES	
Méthodes de validation et d'essai	Z.500–Z.519
INTERGICIELS	
Environnement de traitement réparti	Z.600–Z.609

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

Recommandation UIT-T Z.100

SDL: langage de description et de spécification

Corrigendum 1

Résumé

Le présent corrigendum indique un certain nombre de modifications de forme de la Rec. UIT-T Z.100 (08/2002) ainsi que de l'Amendement 1/Z.100 (10/2003) visant à corriger et à clarifier certains points d'importance secondaire du texte même de la Rec. UIT-T Z.100 et de son Annexe B.

Les modifications se divisent en trois catégories:

- les modifications à apporter au texte même (et à l'Annexe D) de la Rec. UIT-T Z.100 (08/2002);
- la modification du § B.9 de l'Annexe B;
- les adjonctions à l'Annexe B.

Chacune des modifications à apporter au texte même (et à l'Annexe D) de la Rec. UIT-T Z.100 est précédée d'un intitulé comportant successivement un numéro de paragraphe, l'indication du type de modification à apporter (clarification, correction d'une défaillance, extension ou modification – voir l'Appendice II/Z.100), ainsi que le numéro du paragraphe du texte de la Rec. UIT-T Z.100 à modifier. Dans la plupart des cas, la subdivision (*Grammaire abstraite*, *Grammaire concrète*, *Sémantique* ou *Modèle*) du texte de la Rec. UIT-T Z.100 est également indiquée. Enfin, l'intitulé de chaque paragraphe contient quelques mots d'explication sur la nature de la modification.

Les modifications à apporter à l'Annexe B visent à ajouter la syntaxe existante pour les définitions des types de données, qui avaient été omises dans la première publication de cette annexe dans l'Amendement 1 de la Rec. UIT-T Z.100 (10/2003).

Source

Le Corrigendum 1 de la Recommandation UIT-T Z.100 (2002) a été approuvé le 29 août 2004 par la Commission d'études 17 (2001-2004) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8.

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.

L'Assemblée mondiale de normalisation des télécommunications (AMNT), qui se réunit tous les quatre ans, détermine les thèmes d'étude 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 1 de l'AMNT.

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.

Le respect de cette Recommandation se fait à titre volontaire. Cependant, il se peut que la Recommandation contienne certaines dispositions obligatoires (pour assurer, par exemple, l'interopérabilité et l'applicabilité) et considère que la Recommandation est respectée lorsque toutes ces dispositions sont observées. Le futur d'obligation et les autres moyens d'expression de l'obligation comme le verbe "devoir" ainsi que leurs formes négatives servent à énoncer des prescriptions. L'utilisation de ces formes ne signifie pas qu'il est obligatoire de respecter la Recommandation.

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 2005

Tous droits réservés. Aucune partie de cette publication ne peut être reproduite, par quelque procédé que ce soit, sans l'accord écrit préalable de l'UIT.

TABLE DES MATIÈRES

	Page
1 Modifications à apporter au texte même de la Rec. UIT-T Z.100 (et à son Annexe D).....	1
1.1 Extension – 6.6 – Grammaire concrète – Permet de vider la zone <page number area>	1
1.2 Correction d'une défaillance – 7.2 – Grammaire concrète –Remplacement de "connected" par "attached"	1
1.3 Correction d'une défaillance – 8.1.1.1 – Grammaire concrète – Syntaxe incorrecte	1
1.4 Clarification – 8.1.1.1 – Modèle – Virtualité des états composites implicites	1
1.5 Correction d'une défaillance – 8.1.1.5 – Grammaire concrète – Syntaxe incorrecte	2
1.6 Correction d'une défaillance – 8.1.1.5 – Grammaire concrète – Syntaxe incorrecte	2
1.7 Correction d'une défaillance – 8.1.5 – Grammaire concrète – Restriction inutile.....	2
1.8 Correction d'une défaillance – 8.3.1 – Grammaire concrète – Remplacement de "connected" par "attached"	2
1.9 Correction d'une défaillance – 8.5 – Grammaire concrète –Remplacement de "connected" par "attached"	2
1.10 Correction d'une défaillance – 9 – Grammaire concrète – Syntaxe incorrecte	3
1.11 Correction d'une défaillance – 9 – Grammaire concrète – Remplacement de "connected" par "attached"	3
1.12 Correction d'une clarification/défaillance – 9 – Sémantique – Absence d'automate à états.....	3
1.13 Correction d'une défaillance – 9.2 – Grammaire concrète – Syntaxe incorrecte	3
1.14 Correction d'une défaillance – 9.3 – Grammaire concrète – Syntaxe incorrecte	3
1.15 Clarification – 9.5 – Grammaire concrète – Syntaxe incorrecte	4
1.16 Correction d'une défaillance – 10.1, 10.2 – Grammaire concrète – Remplacement de "connected" par "attached"	4
1.17 Clarification – 11.2 – Grammaire concrète – Syntaxe incorrecte	5
1.18 Clarification – 11.2 – Grammaire concrète – Utilisation du nom <composite state name>	5
1.19 Clarification – 11.2 – Modèle – Utilisation du nom <composite state name>	5
1.20 Correction d'une défaillance – 11.3 – Grammaire concrète – Syntaxe incorrecte	5
1.21 Correction d'une défaillance – 11.6 – Grammaire concrète – Syntaxe incorrecte	5

	Page
1.22 Correction d'une défaillance – 11.9 – Grammaire concrète – Syntaxe incorrecte	6
1.23 Clarification – 11.11 – Utilisation du nom <composite state name>	6
1.24 Clarification – 11.11 – Grammaire abstraite – "Procédure à plusieurs états" pour procédures d'entrée/de sortie	6
1.25 Correction d'une défaillance – 11.11.1 – Grammaire concrète – Syntaxe incorrecte	6
1.26 Clarification – 11.11.1 – Grammaire concrète – Utilisation du nom <composite state name>	7
1.27 Clarification – 11.11.1 – Grammaire concrète – Syntaxe incorrecte	7
1.28 Correction d'une défaillance – 11.11.2 – Grammaire concrète – Syntaxe incorrecte	7
1.29 Clarification – 11.11.2 – Grammaire concrète – Utilisation du nom <composite state name>	7
1.30 Correction d'une défaillance – 11.11.2 – Grammaire concrète – Autoriser la présence d'accès dans la définition <inherited state partition definition> ..	7
1.31 Correction d'une défaillance – 11.11.2 – Grammaire concrète – Syntaxe incorrecte	7
1.32 Correction d'une défaillance – 11.11.3 – Grammaire concrète – Syntaxe incorrecte	8
1.33 Clarification – 11.11.4 – Grammaire concrète – Syntaxe incorrecte	8
1.34 Clarification – 11.12.2.2 – Grammaire concrète – Syntaxe incorrecte	8
1.35 Clarification – 11.13.3 – Grammaire concrète – Ambiguïté	8
1.36 Modification – 11.13.4 – Sémantique – Routage des signaux	9
1.37 Modification – 12.1.2 – Modèle d'interfaces implicites	9
1.38 Correction d'une défaillance – 12.1.8 – Grammaire concrète – Syntaxe incorrecte	10
1.39 Clarification – 12.3.4.1 – Sémantique – Unité de temps	10
1.40 Clarification – 13.1 – Grammaire concrète – Syntaxe améliorée	11
1.41 Clarification – D.3.11.2, D.3.12.2 – Unité de temps	11
2 Modification du § B.9 (Annexe B – Compatibilité en amont)	11
B.9 Comportement des opérations	11
3 Adjonctions à l'Annexe B – Compatibilité en amont	12
B.11 Définition des données	12
B.12 Définition des types de données	12
B.13 Types synonymes (<i>syntype</i>)	14

Recommandation UIT-T Z.100

SDL: langage de description et de spécification

Corrigendum 1

1 Modifications à apporter au texte même de la Rec. UIT-T Z.100 (et à son Annexe D)

L'intitulé de chacun des paragraphes ci-dessous du présent paragraphe comporte:

- un numéro de paragraphe (1.1, par exemple);
- l'indication du type de modification à apporter (clarification, correction d'une défaillance, extension ou modification);
- le numéro du paragraphe à modifier;
- à titre facultatif, la subdivision du texte dudit paragraphe (*Grammaire abstraite*, *Grammaire concrète*, *Sémantique* ou *Modèle*);
- quelques mots d'explication sur la nature de la modification.

1.1 Extension – 6.6 – Grammaire concrète – Permet de vider la zone <page number area>

Dans la *Grammaire concrète*, insérer un nouveau crochet "[" juste devant <implicit text symbol> et supprimer le crochet "[" devant <page number> de manière à obtenir la règle suivante:

[<implicit text symbol> *contains* <page number> [(<number of pages>)]]

1.2 Correction d'une défaillance – 7.2 – Grammaire concrète – Remplacement de "connected" par "attached"

Dans la règle <package dependency area>, remplacer "*connected*" par "*attached*" du fait que le diagramme <package diagram> ou la zone <package reference area> est produit(e) par la règle relative au diagramme englobant.

1.3 Correction d'une défaillance – 8.1.1.1 – Grammaire concrète – Syntaxe incorrecte

Dans la règle <agent type diagram>, insérer "{" devant <system type diagram>, insérer "}" après <process type diagram> et insérer "]" après <package use area> de manière à obtenir la règle suivante:

<agent type diagram> ::=
 { <system type diagram> | <block type diagram> | <process type diagram> }
 [*is associated with* <package use area>]

1.4 Clarification – 8.1.1.1 – Modèle – Virtualité des états composites implicites

Pour bien préciser qu'un automate à états implicites d'un type d'agent peut être redéfini si le type d'agent peut être redéfini, les modifications suivantes sont apportées au texte.

Dans le § 8.1.1.1 sous Modèle

A la fin de l'alinéa se finissant par "représenté par la zone <agent body area>.", insérer la phrase suivante: "La virtualité de la définition d'état composite est identique à la virtualité du type d'agent."

Dans l'alinéa qui suit le précédent, au-dessous des deux lignes commençant par un tiret, remplacer le membre de phrase "type d'état composite, implicite et virtuel." par "type d'état composite implicite ayant la même virtualité que le type d'agent."

1.5 Correction d'une défaillance – 8.1.1.5 – Grammaire concrète – Syntaxe incorrecte

Dans la règle <composite state type diagram> *insérer* l'accolade manquante "}" après l'élément <composite state structure area>.

1.6 Correction d'une défaillance – 8.1.1.5 – Grammaire concrète – Syntaxe incorrecte

Dans la règle <composite state type diagram> *remplacer* "**associated with**" devant { <state connection point>* } *set* par "**connected to**" du fait que les éléments sont liés (comme des accès) et que le texte les décrit également comme étant "reliés" (connected). Par ailleurs, remplacer toutes les occurrences de <state connection point> dans la Rec. UIT-T Z.100 *par* <state connection point area> par souci de concordance avec la méta-grammaire graphique.

1.7 Correction d'une défaillance – 8.1.5 – Grammaire concrète – Restriction inutile

Supprimer le dernier alinéa de ce paragraphe, juste avant Sémantique, ainsi libellé:

"L'identificateur <interface identifier> d'une définition <interface gate definition> ne doit pas désigner l'interface implicitement définie par l'entité à laquelle l'accès est connecté (voir § 12.1.2)."

1.8 Correction d'une défaillance – 8.3.1 – Grammaire concrète – Remplacement de "connected" par "attached"

Dans la règle <specialization area> *remplacer* "connected" *par* "attached" du fait que la zone <type reference area> est produite par la règle relative au diagramme englobant.

1.9 Correction d'une défaillance – 8.5 – Grammaire concrète – Remplacement de "connected" par "attached"

Il fallait que la zone <association area> soit "rattachée" à la zone <linked type reference area> du fait que la zone de référence est produite par la règle relative au diagramme englobant.

Remplacer la règle <association area> *par* la suivante:

```
<association area> ::=  
    <association symbol>  
    [ is associated with <association name> ]  
    is attached to { <linked type reference area> <linked type reference area> } set  
    is associated with { <association end area> <association end area> } set
```

et *ajouter* après cette règle l'alinéa suivant:

"Les zones <linked type reference area> sont rattachées à l'une ou l'autre extrémité du symbole <association symbol> et la zone <association end area> la plus proche de l'extrémité correspondante du symbole <association symbol> est à prendre en considération pour cette zone <linked type reference area>."

Remplacer la règle <association end area> existante *par* la suivante:

```
<association end area> ::=  
    { [<role name>] [<multiplicity>] [<ordering area>] [<symbolic visibility>] } set
```

et, dans les alinéas qui suivent la syntaxe, *apporter* les modifications suivantes:

Remplacer "Si une zone <association end area> identifie" *par* "Si une zone <association end area> correspond à".

Remplacer "est reliée à une zone <linked type reference area>" *par* "est rattachée à une zone <linked type reference area>".

1.10 Correction d'une défaillance – 9 – Grammaire concrète – Syntaxe incorrecte

Dans la règle <agent body area>, la zone facultative <on exception association area> devrait être indépendante de toute zone <start area>.

Supprimer "[" devant [<on exception association area> et insérer "[" devant <start area> de manière à obtenir la ligne de syntaxe suivante:

{ [<on exception association area>] [<start area>]

1.11 Correction d'une défaillance – 9 – Grammaire concrète – Remplacement de "connected" par "attached"

Dans la règle <create line area>, remplacer "**connected**" par "**attached**" du fait que l' (le type d') agent ou la partition d'état est produit(e) par la règle relative au diagramme englobant.

1.12 Correction d'une clarification/défaillance – 9 – Sémantique – Absence d'automate à états

Le but était de définir de manière plus claire l'automate à états dans le cas où il n'y a pas d'autre automate à états, explicite ou implicite. La première partie de la deuxième phrase "Un agent englobant des instances initiales" aurait dû être mis à la forme négative "Un agent n'englobant pas d'instances initiales".

L'alinéa:

"Si un agent n'a pas d'automate à états explicite ou implicite, dès que tous les agents contenus initialement ont été créés, cet agent entre dans un état de mise à l'arrêt. Un agent englobant des instances initiales et ne contenant pas d'automates à états cesse donc d'exister dès qu'il est créé."

Est *modifié* comme suit:

"Si un agent n'a pas d'automate à états explicite ou implicite, cela révèle l'existence d'un automate à états qui ne comporte qu'un seul nœud d'arrêt *Stop-node*. Dès que tous les agents contenus initialement ont été créés, cet agent entre dans un état de mise à l'arrêt. Un agent n'englobant pas d'instances initiales et ne contenant pas d'automates à états cesse donc d'exister dès qu'il est créé."

1.13 Correction d'une défaillance – 9.2 – Grammaire concrète – Syntaxe incorrecte

Dans la règle <block diagram>, les identificateurs <external channel identifiers> sont associés avec (**associated with**) le bloc (et non pas reliés ("**connected to**") à celui-ci). La ligne de syntaxe:

is connected to { {<gate on diagram> | <external channel identifiers>}* } **set**

est *remplacée* par:

is connected to { {<gate on diagram> }* } **set**
is associated with { <external channel identifiers> }* } **set**

1.14 Correction d'une défaillance – 9.3 – Grammaire concrète – Syntaxe incorrecte

Dans la règle <process diagram>, les identificateurs <external channel identifiers> sont associés avec (**associated with**) le processus (et non pas reliés ("**connected to**") à celui-ci). La ligne de syntaxe:

is connected to { {<gate on diagram> | <external channel identifiers>}* } **set**

est *remplacée* par:

is connected to { {<gate on diagram> }* } **set**
is associated with { <external channel identifiers> }* } **set**

1.15 Clarification – 9.5 – Grammaire concrète – Syntaxe incorrecte

Dans la règle <procedure body area>, *mettre* la règle entre accolades {} et ajouter *set* à droite de cette règle pour indiquer clairement que l'ordre est sans importance. Les lignes de syntaxe ainsi obtenues sont les suivantes:

```
{[ <on exception association area> ] [<procedure start area>]
  {<state area> | <exception handler area> | <in connector area> } * } set
```

1.16 Correction d'une défaillance – 10.1, 10.2 – Grammaire concrète – Remplacement de "connected" par "attached"

La règle <channel definition area> est reliée (*connected to*) à des agents, des états ou des accès à l'une et l'autre extrémité du canal. Toutefois, *connected to* signifie que l'élément syntaxique est produit et qu'il est par conséquent incorrect. Le métasymbole correct est *attached to*. Le mot "*connected*" est remplacé par "*attached*" dans la syntaxe. La syntaxe révisée s'établit comme suit:

```
<channel definition area> ::=
    <channel symbol>
    is associated with
    { [<channel name>] { [<signal list area>] [<signal list area>] } set }
    is attached to {
    { <agent area> | <state partition area> | <gate on diagram> }
    { <agent area> | <state partition area> | <gate on diagram> } } set
```

En conséquence, *remplacer* "connecté", "connectés", "connectée" ou "connectées" *par* "rattaché", "rattachés", "rattachée" ou "rattachées" dans le reste du texte du § 10.1 (13 fois) et du § 10.2 (5 fois après les autres suppressions indiquées ci-dessous).

Dans le § 10.1 *Grammaire concrète*, *remplacer* également le mot "connexion" par le mot "rattachement" (deux fois).

En procédant à ces modifications, il a été constaté que le texte des sous-titres *Sémantique* et *Modèle* du § 10.2 faisait état de canaux internes à l'unité de portée auxquels les identificateurs <external channel identifi ers> sont rattachés.

En conséquence, *supprimer* la première phrase du deuxième alinéa du sous-titre *Sémantique* du § 10.2 reproduite ci-dessous:

"Chaque canal identifié par un identificateur <channel identifier> contenu dans un identificateur de canal externe <external channel identifi ers> d'une unité de portée donnée doit être défini dans l'agent dans lequel est définie la connexion et doit avoir la frontière de cet agent comme l'une de ses extrémités."

Le passage suivant du § 10.2, sous *Sémantique*, semble constituer une contrainte inutile et il est proposé de le *supprimer*:

"Chaque canal défini dans l'agent qui l'entoure et qui a son environnement comme étant l'une de ses extrémités, doit être mentionné dans exactement un des identificateurs de canal externe <external channel identifi ers>."

Le sous-titre *Sémantique* du § 10.2 est incorrect (le texte a trait essentiellement à la grammaire) et ce sous-titre est *supprimé*.

Le membre de phrase "dans leur unité de portée respective" du § 10.2, sous *Modèle*, est *supprimé* car il est hors de propos – il n'y a qu'une seule unité de portée concernée (l'unité de portée qui l'entoure).

A la fin du texte figurant sous *Modèle* (§ 10.2), *ajouter* la phrase explicative suivante:

"Les identités des canaux et des accès découlent du rattachement (voir le § 10.1)."

1.17 Clarification – 11.2 – Grammaire concrète – Syntaxe incorrecte

Supprimer la règle <connect association area>. Cette règle est définie au § 11.11.4 à la place de la règle <connect area>. Voir aussi la modification apportée au § 11.11.4 au § 1.33 du présent corrigendum.

1.18 Clarification – 11.2 – Grammaire concrète – Utilisation du nom <composite state name>

L'utilisation du nom <composite state name> est améliorée.

Remplacer "Dans ce cas la zone <state area> ne doit contenir qu'un seul nom <composite state name>" par "Dans ce cas, la zone <state area> ne doit contenir qu'un seul nom <composite state name>".

1.19 Clarification – 11.2 – Modèle – Utilisation du nom <composite state name>

L'utilisation du nom <composite state name> est améliorée.

Remplacer "une pour chaque item <state name> et <composite state name> du corps en question" par "une pour chaque item <state name> du corps en question".

1.20 Correction d'une défaillance – 11.3 – Grammaire concrète – Syntaxe incorrecte

La syntaxe correspondant à la zone <input area>:

```
<input symbol> contains { [<virtuality>] <input list> }  
[ is connected to <on exception association area> ]  
[ is associated with <solid association symbol> is connected to <enabling condition area> ]  
is followed by <transition area>
```

signifie que lorsqu'une condition d'activation est utilisée, il devrait y avoir une ligne (un symbole <solid association symbol>) aboutissant à la condition d'activation ET une ligne de liaison aboutissant à la transition en provenance du symbole d'entrée. Cela est manifestement incorrect. La syntaxe corrigée s'établit comme suit:

```
<input symbol> contains { [<virtuality>] <input list> }  
[ is connected to <on exception association area> ]  
{ is connected to <enabling condition association area>  
  | is followed by <transition area> }
```

et est ajoutée après cette syntaxe l'alinéa suivant:

"La zone <enabling condition association area> définit la zone <transition area> dans le cas d'une condition d'activation."

Le § 11.6 est modifié en conséquence (voir le § 1.21 ci-dessous).

1.21 Correction d'une défaillance – 11.6 – Grammaire concrète – Syntaxe incorrecte

Afin de corriger la syntaxe correspondant à la zone <input area> (voir le § 1.20 ci-dessus), une nouvelle règle syntaxique est introduite pour la zone <enabling condition association area> et la règle <enabling condition area> est étendue pour inclure une zone <transition area>. La syntaxe corrigée s'établit comme suit:

```
<enabling condition association area> ::=  
  <solid association symbol> is connected to <enabling condition area>  
<enabling condition area> ::=  
  <enabling condition symbol> contains <provided expression>  
  is followed by <transition area>
```

et est ajoutée après cette syntaxe l'alinéa suivant:

"La zone <transition area> correspond à la *Transition* du nœud *Input-node* ou à la transition *Spontaneous-transition* pour l'expression *Provided-expression*. La syntaxe semble permettre ici d'obtenir la production graphique correcte d'une ligne de liaison entre le symbole <enabling condition symbol> et la transition."

1.22 Correction d'une défaillance – 11.9 – Grammaire concrète – Syntaxe incorrecte

La syntaxe correspondant à la zone <spontaneous transition area> signifie que lorsqu'une condition d'activation est utilisée il devrait y avoir une ligne (un symbole <solid association symbol>) aboutissant à la condition d'activation ET une ligne de liaison aboutissant à la transition en provenance du symbole d'entrée. Cela est manifestement incorrect. La syntaxe corrigée s'établit comme suit:

```
<input symbol> contains { [<virtuality>] <spontaneous designator> }
[ is connected to <on exception association area> ]
{
    | is connected to <enabling condition association area>
    | is followed by <transition area> }
```

et est *ajouté* après cette syntaxe l'alinéa suivant:

"La zone <enabling condition association area> définit la zone <transition area> dans le cas d'une condition d'activation."

Le § 11.6 est modifié en conséquence (voir le § 1.21 ci-dessus).

1.23 Clarification – 11.11 – Utilisation du nom <composite state name>

L'utilisation du nom <composite state name> est améliorée.

Dans le 2^e alinéa, *remplacer* "<composite state name>" *par* "<composite state name>".

1.24 Clarification – 11.11 – Grammaire abstraite – "Procédure à plusieurs états" pour procédures d'entrée/de sortie

Le texte figurant sous la syntaxe abstraite, à partir de "La définition *Entry-procedure-definition* représente" et jusqu'à "*Sémantique*" est *remplacé* par le suivant:

"Une procédure à plusieurs états est une procédure qui contient un état (explicite ou implicite) ou qui appelle une procédure à plusieurs états.

La définition *Entry-procedure-definition* d'un graphe *Composite-state-graph* ou d'un nœud *State-aggregation-node* est une procédure sans paramètres explicitement définis dans le graphe *Composite-state-graph* ou le nœud *State-aggregation-node*, respectivement, avec le nom d'entrée. Une procédure d'entrée ne doit pas être une procédure à plusieurs états.

La définition *Exit-procedure-definition* d'un graphe *Composite-state-graph* ou d'un nœud *State-aggregation-node* est une procédure sans paramètres explicitement définis dans le graphe *Composite-state-graph* ou le nœud *State-aggregation-node*, respectivement, avec le nom de sortie. Une procédure de sortie ne doit pas être une procédure à plusieurs états."

1.25 Correction d'une défaillance – 11.11.1 – Grammaire concrète – Syntaxe incorrecte

Dans la règle <composite state graph area>, *remplacer* "**associated with**" devant { <state connection point>* } *set par* "**connected to**" du fait que les éléments sont liés (comme des accès) et que le texte les décrit également comme étant "reliés ("connected"). Par ailleurs, *renommer* toutes les occurrences de <state connection point> *par* <state connection point area> par souci de concordance avec la grammaire graphique.

1.26 Clarification – 11.11.1 – Grammaire concrète – Utilisation du nom <composite state name>

L'utilisation du nom <composite state name> est améliorée.

Dans la règle <composite state heading>, *remplacer* "<composite state name>" *par* "<state name>".

1.27 Clarification – 11.11.1 – Grammaire concrète – Syntaxe incorrecte

Dans la règle <composite state body area>, *mettre* la syntaxe correspondante entre accolades "{}" et ajouter "**set**" à droite de la syntaxe pour indiquer clairement que l'ordre est sans importance. De plus, la présentation est alignée sur celle des autres règles analogues. Les lignes de syntaxe obtenues sont les suivantes:

```
{ [<on exception association area>] <start area>*  
  { <state area> | <exception handler area> | <in connector area> }* } set
```

1.28 Correction d'une défaillance – 11.11.2 – Grammaire concrète – Syntaxe incorrecte

Dans la règle <state aggregation area> *remplacer* "*associated with*" devant { <state connection point>* } *set par* "**connected to**" du fait que les éléments sont liés (comme des accès) et que le texte les décrit comme étant reliés ("connected"). Par ailleurs, *renommer* toutes les occurrences de <state connection point> *par* <state connection point area> par souci de concordance avec la grammaire graphique.

1.29 Clarification – 11.11.2 – Grammaire concrète – Utilisation du nom <composite state name>

L'utilisation du nom <composite state name> est améliorée.

Dans la règle <state aggregation heading>, *remplacer* "<composite state name>" *par* "<state name>".

1.30 Correction d'une défaillance – 11.11.2 – Grammaire concrète – Autoriser la présence d'accès dans la définition <inherited state partition definition>

Il convient d'autoriser la présence d'accès dans la définition <inherited state partition definition>. La règle révisée s'établit comme suit:

```
<inherited state partition definition> ::=  
  <dashed state symbol> contains { <composite state identifier> { <gate>* } set }
```

1.31 Correction d'une défaillance – 11.11.2 – Grammaire concrète – Syntaxe incorrecte

Par ailleurs, des crochets "[]" à la place d'accolades "{}" ont été utilisés par erreur dans la règle <state partition connection area>. Il ne devrait pas être permis que le symbole <solid association symbol> soit non connecté.

De plus, le point <outer graphical point> a été incorrectement représenté comme étant associé avec un symbole <frame symbol>, alors que le symbole <frame symbol> est une production syntaxique de l'aggrégat d'état englobant (type) (une zone <state aggregation area> ou un diagramme <composite state type diagram>). Le point <outer graphical point> doit être un point <outer graphical point> défini par la zone <state connection point area> du diagramme.

```
<state partition connection area> ::=  
  <solid association symbol>  
  is attached to <frame symbol>  
  is attached to <state partition area>  
  is connected to { <outer graphical point> <inner graphical point> }
```

et *ajouter* l'alinéa suivant:

"Le symbole <solid association symbol> est rattaché à une extrémité au symbole <frame symbol> du diagramme englobant et le point <outer graphical point> est placé à proximité à l'extérieur de ce symbole <frame symbol> du diagramme englobant. Le symbole <solid association symbol> est rattaché à l'autre extrémité à une zone <state partition area> et le point <inner graphical point> est placé à proximité. Le point <outer graphical point> doit se référer uniquement aux noms définis en tant que points d'entrée ou de sortie d'état du diagramme englobant. Le point <inner graphical point> doit se référer uniquement aux noms définis en tant que points d'entrée ou de sortie d'état de la zone <state partition area>."

Supprimer les parties graphiques associées dans les règles relatives aux points graphiques (dont la syntaxe n'est pas conforme à la métagrammaire), qui deviennent:

```
<outer graphical point> ::=
    { <state entry points> | <state exit points> }
<inner graphical point> ::=
    { <state entry points> | <state exit points> }
```

1.32 Correction d'une défaillance – 11.11.3 – Grammaire concrète – Syntaxe incorrecte

Supprimer la dernière ligne "***is connected to*** <frame symbol>" de la règle <state connection point> du fait que c'est le point de connexion qui est relié à la trame et qui produit sur le plan syntaxique le point de connexion (et non pas l'inverse). Par ailleurs, *renommer* toutes les occurrences de <state connection point> *par* <state connection point area> par souci de concordance avec la grammaire graphique.

1.33 Clarification – 11.11.4 – Grammaire concrète – Syntaxe incorrecte

Remplacer la règle <connect area> *par* la suivante:

```
<connect association area> ::=
    <solid association symbol> is associated with { [<virtuality>] [<connect list>] }
    [ is connected to <on exception association area> ]
    is followed by <exit transition area>
```

et *remplacer* toutes les occurrences de <connect area> *par* <connect association area>. Voir aussi la modification apportée au § 11.2 au § 1.17 du présent corrigendum.

1.34 Clarification – 11.12.2.2 – Grammaire concrète – Syntaxe incorrecte

Remplacer ***connected*** *par* ***attached*** dans la règle <merge area> pour faire en sorte qu'une autre production produise la ligne de liaison.

1.35 Clarification – 11.13.3 – Grammaire concrète – Ambiguïté

Dans la règle <procedure call body>, les options <procedure identifier> et <procedure type expression>, étaient ambiguës car un identificateur <procedure identifier> est une expression <procedure type expression>. La syntaxe révisée s'établit comme suit:

```
<procedure call body> ::=
    [ this ] <procedure type expression> [<actual parameters>]
```

Après cette syntaxe, l'alinéa suivant est *ajouté* pour expliquer ce que désigne l'identificateur <procedure identifier> dans le reste du texte de ce paragraphe.

"Dans le texte qui suit, le terme <procedure identifier> désigne l'identificateur <procedure identifier> du type <base type> de l'expression <procedure type expression>, et si l'expression <procedure type expression> est un identificateur <procedure identifier>, il s'agit simplement de l'identificateur <procedure identifier> considéré ici."

1.36 Modification – 11.13.4 – Sémantique – Routage des signaux

Pour pouvoir envoyer un signal à une instance d'agent en faisant référence à son Pid et autoriser cet agent à router par voie interne le signal à destination de l'agent contenu voulu, l'interface d'un agent doit comporter une voie de communication à destination des agents contenus. Cela rend également nécessaire la modification du § 1.37 ci-dessous.

L'alinéa suivant est inséré avant l'alinéa commençant par "Il faut noter que spécifier le même identificateur *Channel-identifier*":

"Lorsqu'une instance du signal est remise à une instance d'un ensemble d'instances d'agent et qu'il existe un trajet de communication interne qui achemine le signal à destination de l'automate à états de l'instance d'agent, l'instance du signal est remise à cet automate à états. Si tel n'est pas le cas, un trajet de communication interne à l'instance d'agent à même d'acheminer l'instance du signal est choisi de manière arbitraire et l'instance du signal est remise à un ensemble d'instances d'un agent contenu."

1.37 Modification – 12.1.2 – Modèle d'interfaces implicites

Pour pouvoir envoyer un signal à une instance d'agent en faisant référence à son Pid et autoriser cet agent à router par voie interne le signal à destination de l'agent contenu voulu, l'interface d'un agent doit comporter une voie de communication à destination d'agents contenus. Le texte définissant l'interface implicite étant difficile à comprendre, on a cherché à l'améliorer.

L'alinéa:

"L'interface définie par un agent ou un type d'agent contient dans sa spécialisation <interface specialization> toutes les interfaces figurant dans la liste de signaux en entrée associée avec un accès explicite ou implicite pour l'agent ou le type d'agent, de manière à ce que ces accès soient connectés par des canaux explicites ou implicites vers un accès de la machine d'état de l'agent ou du type d'agent. L'interface contient également dans sa liste <interface use list> tous les signaux, variables distantes et procédures distantes figurant dans la liste de signaux en entrée associée avec un accès explicite ou implicite pour l'agent ou le type d'agent, de manière à ce que ces accès soient connectés par des canaux explicites ou implicites vers un accès de la machine d'état de l'agent ou du type d'agent. En outre, l'interface pour un type d'agent qui hérite d'un autre type d'agent contient également dans sa spécialisation <interface specialization> l'interface implicite définie par le type d'agent hérité."

est *remplacé* par l'alinéa suivant:

"Les accès d'un agent (ou d'un type d'agent) connectés par voie interne sont des accès explicites ou implicites de l'agent (ou, respectivement, du type d'agent) qui sont connectés via des canaux implicites ou explicites aux accès soit de l'automate à états de l'agent (ou, respectivement, du type d'agent) soit d'un agent contenu. L'interface définie par un agent ou un type d'agent contient dans sa spécialisation <interface specialization> toutes les interfaces figurant dans la liste de signaux en entrée associée avec des accès connectés par voie interne. L'interface contient dans sa liste <interface use list> tous les signaux, variables distantes et procédures distantes figurant dans la liste de signaux en entrée associée avec des accès connectés par voie interne. En outre, l'interface pour un type d'agent qui hérite d'un autre type d'agent contient également dans sa spécialisation <interface specialization> l'interface implicite définie par le type d'agent hérité."

Dans le texte de cet alinéa, le membre de phrase "soit de l'automate à états de l'agent (ou, respectivement, du type d'agent) soit d'un agent contenu" est remplacé par "de l'automate à états de l'agent (ou, respectivement, du type d'agent)", formulation qui devrait avoir le même sens que la précédente.

L'alinéa suivant, qui définit l'interface implicite de l'automate à états d'un agent, doit donc être modifié pour en exclure les éléments ne concernant que les agents contenus. On s'est également

efforcé d'améliorer ce texte – en particulier la description des éléments ne concernant pas la communication via les accès des agents. Le texte initial:

"L'interface définie par la machine d'état d'un agent ou d'un type d'agent contient dans sa spécialisation <interface specialization> l'interface définie par l'agent ou le type d'agent. Elle contient en outre dans sa spécialisation <interface specialization> toutes les interfaces figurant dans la liste de signaux en entrée associée avec un accès explicite ou implicite de la machine d'état de sorte que les accès ne sont pas connectés par des canaux explicites ou implicites à un accès explicite ou implicite pour l'agent ou le type d'agent. L'interface contient également dans sa liste <interface use list> tous les signaux, variables distantes et procédures distantes figurant dans la liste de signaux en entrée associée avec un accès explicite ou implicite de la machine d'état de sorte que les accès ne sont pas connectés par des canaux explicites ou implicites à un accès explicite ou implicite de l'agent ou du type d'agent. Si l'entité qui l'englobe est un type d'agent qui hérite d'un autre type d'agent, l'interface contiendra alors également dans sa spécialisation <interface specialization> l'interface implicite de la machine d'état du type d'agent hérité."

est *remplacé* par le texte suivant:

"L'interface définie par l'automate à états d'un agent ou d'un type d'agent contient dans sa spécialisation <interface specialization> l'interface définie par l'agent ou le type d'agent, à l'exclusion de toute partie de cette interface ne concernant que les agents contenus. Toutefois, l'interface contient en outre dans sa spécialisation <interface specialization> toutes les interfaces figurant dans la liste de signaux en entrée associée avec un accès explicite ou implicite de l'automate à états. L'interface contient également dans sa liste <interface use list> tous les signaux, variables distantes et procédures distantes figurant dans la liste de signaux en entrée associée avec un accès explicite ou implicite de l'automate à états. Si l'entité qui englobe est un type d'agent qui hérite d'un autre type d'agent, l'interface contiendra alors également dans sa spécialisation <interface specialization> l'interface implicite de l'automate à états du type d'agent hérité."

On peut modifier ce texte pour lui donner le même sens que le texte initial en supprimant le membre de phrase "à l'exclusion de toute partie de cette interface ne concernant que les agents contenus".

1.38 Correction d'une défaillance – 12.1.8 – Grammaire concrète – Syntaxe incorrecte

Dans la règle <operation body area>, la zone <procedure start area> devrait être facultative. Les crochets "[" et "]" sont ajoutés respectivement devant et après <procedure start area>. On obtient ainsi une définition améliorée ou abstraite permettant d'omettre la zone <procedure start area>. La syntaxe correspondante est mise entre accolades "{}" et "*set*" est *ajouté* à droite de la syntaxe pour indiquer clairement que l'ordre est sans importance. Les lignes de syntaxe révisées sont les suivantes:

```
{ [ <on exception association area> ] [ <procedure start area> ]  
{ <in connector area> | <exception handler area> } * } set
```

1.39 Clarification – 12.3.4.1 – Sémantique – Unité de temps

La phrase suivante:

"Sauf indication contraire, l'unité de temps des spécifications SDL est de 1 seconde."

est *ajoutée* aux emplacements suivants:

- Dans le § 12.3.4.1, sous *Sémantique*, après le membre de phrase "se présentent dans la même transition".
- Dans le § D.3.11.2 Utilisation (relatif à la sorte Durée (*Duration*)), à la fin du paragraphe, juste avant le § D.3.12.
- Dans le § D.3.12.2 Utilisation (relatif à la sorte Temps (*Time*)), à la fin du paragraphe, juste avant le § D.3.13.

1.40 Clarification – 13.1 – Grammaire concrète – Syntaxe améliorée

La règle <option symbol> est *redéfinie* comme suit:

```
<option symbol> ::=
    {
        <dashed line symbol> is attached to <dashed line symbol>
        <dashed line symbol> is attached to <dashed line symbol>
        <dashed line symbol> is attached to <dashed line symbol>
        { <dashed line symbol> is attached to <dashed line symbol> }+ } set
```

et le mot "rectiligne" est *inséré* après le mot "polygone" dans la phrase qui suit la règle.

1.41 Clarification – D.3.11.2, D.3.12.2 – Unité de temps

Voir le § 1.39 du présent corrigendum.

2 Modification du § B.9 (Annexe B – Compatibilité en amont)

Certains éléments grammaticaux nécessaires à la prise en charge d'opérateurs définis selon la syntaxe employant le langage SDL-92 manquaient dans l'édition précédente. Le texte qui suit remplace le § B.9 figurant dans l'Amendement 1 à la Z.100 (10/2003) et incorpore le texte précédent.

B.9 Comportement des opérations

Grammaire concrète

Pour assurer la compatibilité avec les modèles employant le langage SDL-92, les nouvelles unités syntaxiques suivantes sont ajoutées pour les définitions d'opérateur: <legacy operator definition>, <legacy operator reference> et <legacy external operator definition>. Ces unités sont utilisées à la place de l'unité <operation definitions> dans une unité <legacy data type definition> (voir le § B.12).

```
<legacy operator definition> ::=
    {<package use clause>}*
    <operation heading> <end>
    { <entity in operation> }*
    <start>
    endoperator
    [ {<operation identifier> | <operation name> } ] <end>
<legacy operator reference> ::=
    <operation heading> referenced <end>
<legacy external operator definition> ::=
    operator <operation name> [ <legacy procedure signature> ] external <end>
```

Une unité <operation heading> dans une unité <legacy operator definition> ou une unité <legacy operator reference> doit utiliser le mot clé **operator**.

Une unité <legacy operator definition> correspond à une unité <operation definition> en langage SDL-2000.

Une unité <legacy operator reference> correspond à une unité <operation reference> en langage SDL-2000.

Une unité <legacy external operator definition> correspond à une unité <external operation definition> en langage SDL-2000.

L'unité <start> est définie dans la Rec. UIT-T Z.106. Le corps de la transmission de l'unité <start> ne doit contenir que des éléments qui sont autorisés dans la définition d'une opération.

La syntaxe pour l'unité <formal operation parameters> est étendue de manière que soient compris les paramètres formels spécifiés par le mot clé **fpar**.

```

<formal operation parameters> ::=
    ( <operation parameters> {, <operation parameters> }* )
    |   [<end>] fpar <operation parameters> {, <operation parameters> }*

```

NOTE – L'unité facultative <end> avant le mot clé **fpar** est ajoutée en signe d'agrément des modèles à définir au moyen des outils qui nécessitent à ce stade un point-virgule, même si elle n'est pas valable en SDL/GR dans le langage SDL-92.

La syntaxe pour l'unité <operation result> est étendue de manière à permettre la spécification avec le mot clé **returns**.

```

<operation result> ::=
    <result sign> [<variable name>] <sort>
    |   returns [ <variable name> ] <sort>

```

3 Adjonctions à l'Annexe B – Compatibilité en amont

La grammaire définie dans l'Annexe B de l'Amendement 1 à la Z.100 (10/2003) ne contenait pas certains des éléments grammaticaux nécessaires à la prise en charge des données et des types de données utilisant la syntaxe employant le langage SDL-92. Les points suivants sont *ajoutés* à l'Annexe B pour permettre la prise en charge de cette syntaxe employant le langage SDL-92.

B.11 Définition des données

Grammaire concrète

Pour être compatible avec les modèles employant le langage SDL-92, la syntaxe est étendue de manière que soient comprises les unités <legacy data type definition> (voir le § B.12) et <legacy syntype definition> (voir le § B.13) dans une unité <data definition>.

```

<data definition> ::=
    <data type definition>
    |   <legacy data type definition>
    |   <interface definition>
    |   <syntype definition>
    |   <legacy syntype definition>
    |   <synonym definition>

```

Pour être conforme au langage SDL-92, une unité <sort> de l'un quelconque des composants d'une unité <data definition> devrait toujours être un identificateur de sorte ou de type synonyme (*syntype*).

B.12 Définition des types de données

Grammaire concrète

Pour être compatible avec les modèles employant le langage SDL-92 la syntaxe est étendue de manière que soit comprise l'unité <legacy syntype definition> (voir le § B.13) dans l'unité <entity in data type>.

```

<entity in data type> ::=
    <data type definition>
    |   <legacy data type definition>
    |   <syntype definition>
    |   <legacy syntype definition>
    |   <synonym definition>
    |   <exception definition>
<legacy data type definition> ::=
    newtype <sort name>
        [<formal context parameters>]
        [<data type specialization>]
        [<legacy generators>]
        [<structure definition>]

```

```

[<literal list> ]
[<legacy operator signatures> ]
{
  <legacy operator definition>
  | <legacy operator reference>
  | <legacy external operator definition> }*
[ <default initialization> [ <end> ] ]
[ constants <range condition> ]
endnewtype [ <sort name> ]

```

Pour être conforme au langage SDL-92, l'unité <data type specialization> d'une unité <legacy data type definition> devrait contenir une unité <legacy data inheritance> (voir le § B.8).

Pour être conforme au langage SDL-92, l'unité <structure definition> d'une unité <legacy data type definition> ne devrait pas contenir l'unité <visibility>, le mot clé **optional** ou l'unité <field default initialisation>.

Pour être conforme au langage SDL-92, l'unité <literal list> d'une unité <legacy data type definition> ne devrait pas contenir l'unité <visibility> ni l'unité <named number>.

Si une définition <legacy data type definition> contient une unité <range condition>, elle représente la définition du type synonyme (*syntype*) et d'un type de données parentes anonymes.

Un opérateur existant doit être défini par l'unité <legacy operator definition> ou référencé par une unité <legacy operator reference> ou une unité <legacy external operator definition> (voir le § B.9).

B.12.1 Générateurs

Grammaire concrète

Bien que le langage SDL-2000 ne contienne pas de générateurs pour les types de données ni les types de données paramétrées dans le paquetage **package** Predefined du langage SDL-2000, remplacer les générateurs tels que Array (Tableau) qui figuraient dans le paquetage **package** Predefined dans le langage SDL-92. L'unité <legacy data type definition> incluant la syntaxe <legacy generators>, ces types de données paramétrées peuvent être utilisées.

```

<legacy generators> ::=
    <sort identifier> (<legacy generator actual> { , <legacy generator actual> }* )
<legacy generator actual> ::=
    <sort>
    | <literal signature>
    | <operator name>
    | <constant expression>

```

L'unité <sort identifier> devrait identifier un des types de données paramétrées dans le paquetage **package** Predefined. L'unité <legacy generator actual> devrait constituer un paramètre effectif approprié pour les types de données paramétrées.

B.12.2 Signatures d'opérateur

Grammaire concrète

Pour être compatible avec les modèles employant le langage SDL-92, l'unité <legacy operator signatures> est autorisée à titre d'unité alternative.

```

<legacy operator signatures> ::=
    operators
    <legacy operator signature> { <end> <legacy operator signature> }* [ <end> ]
<legacy operator signature> ::=
    <operator name> : <arguments> -> <sort>

```

L'unité <legacy operator signature> représente une signature *Operation-Signature*.

L'unité <sort> d'une unité <legacy operator signature> représente le résultat *Result* de la signature *Operation-Signature*.

B.13 Types synonymes (*syntype*)

Grammaire concrète

Pour être compatible avec les modèles employant le langage SDL-92, l'unité <legacy syntype definition> est autorisée à titre d'unité alternative.

<legacy syntype definition> ::=

```
    syntype
      <syntype name> = <parent sort identifier>
      [ <default initialization> [ <end> ] ]
      [ constants <range condition> ]
    endsyntype [ <syntype name> ]
```

Voir aussi l'unité <legacy data type definition> pour le mot clé **syntype** combiné avec un mot clé **newtype** au § B.12.

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
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	Réseaux câblés et 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	Gestion des télécommunications y compris le RGT et maintenance des réseaux
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 de données, communication entre systèmes ouverts et sécurité
Série Y	Infrastructure mondiale de l'information, protocole Internet et réseaux de nouvelle génération
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication