UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Addendum 1
Q.1400

(02/95)

RÉSEAU INTELLIGENT

CADRE ARCHITECTURAL D'ÉLABORATION DES PROTOCOLES DE SIGNALISATION ET D'EXPLOITATION, ADMINISTRATION ET MAINTENANCE UTILISANT LES CONCEPTS DE L'INTERCONNEXION DES SYSTÈMES OUVERTS

Addendum 1 à la
Recommandation UIT-T Q.1400

(Antérieurement «Recommandation du CCITT»)

AVANT-PROPOS

L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'Union internationale des télécommunications (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 (Helsinki, 1er-12 mars 1993).

L'Addendum 1 à la Recommandation UIT-T Q.1400, que l'on doit à la Commission d'études 11 (1993-1996) de l'UIT-T, a été approuvé le 7 février 1995 selon la procédure définie dans la Résolution n° 1 de la CMNT.

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 de télécommunications.

Addendum 1 à la Recommandation Q.1400

CADRE ARCHITECTURAL D'ÉLABORATION DES PROTOCOLES DE SIGNALISATION ET D'EXPLOITATION, ADMINISTRATION ET MAINTENANCE UTILISANT LES CONCEPTS DE L'INTERCONNEXION DES SYSTÈMES OUVERTS

(Genève, 1995)

De nouvelles règles de compatibilité en aval et en amont pour les protocoles spécifiés en ASN.1 ont été adoptées. Ces règles sont simples à utiliser et sont fondées sur les plus récentes Recommandations de la série X.680 relatives à l'ASN.1. Le paragraphe 12.5/Q.1400 (Recommandation publiée en 1993) doit être remplacé par le texte suivant.

12.5 Règles de compatibilité pour les protocoles de couche application spécifiés en ASN.1

La possibilité qu'un protocole d'application fasse de temps en temps l'objet d'extensions mineures a été envisagée. Une syntaxe abstraite est étendue si le type qui lui est associé l'est aussi (autrement dit, s'il s'agit d'un type choice, il est possible d'étendre cette syntaxe en lui ajoutant une nouvelle composante ou en étendant une composante existante). Une manière d'étendre une PDU (ou tout type de structure) consiste à étendre le type de l'une quelconque de ses composantes. Lorsqu'on prend en charge de telles extensions, des précautions doivent être prises pour que lesdites extensions soient effectivement mineures. Par conséquent, les types suivants d'extension de la syntaxe abstraite pourraient être considérés comme mineurs:

- ajout d'un élément d'information qui améliorera peut-être une activité mais n'est pas essentiel pour l'exécution de l'activité de base (par exemple liste des options supplémentaires d'acheminement); ou
- ajout d'un élément d'information pour introduire une possibilité qui n'est pas essentielle à la base (par exemple, ajout du «nom» au «numéro» pour qu'il s'affiche sur l'écran du terminal).

Dans ces cas mentionnés ci-dessus, il n'est pas nécessaire de définir un nouveau nom de contexte d'application. Toutefois, les procédures de compatibilité aval relatives au traitement des informations non connues doivent exister dans le processus d'application récepteur.

Les types suivants d'extension peuvent être considérés comme majeurs:

- ajout d'une nouvelle procédure; ou
- modification fondamentale d'une procédure (par exemple «exécuter cette procédure deux fois»).

Dans ce cas, un nouveau nom de contexte d'application doit être défini.

Les règles de compatibilité en amont et en aval suivantes sont appliquées aux protocoles de couche application spécifiés à l'aide de la notation ASN.1.

12.5.1 Règles de compatibilité en amont

Les modifications de toutes les versions futures des Recommandations UIT-T sur les protocoles de signalisation fondés sur l'ASN.1 doivent être effectuées, à partir de 1992, de manière à assurer la compatibilité en amont, c'est-à-dire avec les anciennes versions.

L'objectif des paragraphes qui suivent est de donner des directives sur les types de modification que l'on peut apporter à une spécification de protocole tout en assurant la compatibilité en amont avec le protocole initial.

Les modifications qui n'influent pas sur la syntaxe de transfert (c'est-à-dire les bits et les octets échangés entre des entités homologues) ou qui l'étendent sont compatibles en amont. En termes simples, la compatibilité en amont signifie qu'une PDU codée du protocole initial est une PDU codée valable pour le nouveau protocole.

En dehors de cas très particuliers, les modifications qui n'influent pas sur la syntaxe abstraite ou qui l'étendent produisent une syntaxe de transfert compatible en amont lorsqu'on utilise des BER. Ces modifications sont énumérées aux 12.5.1.1 et 12.5.1.2.

Cependant, en l'absence de règles de compatibilité en amont, un interfonctionnement correct ne peut être assuré que si les valeurs étendues (c'est-à-dire les valeurs qui appartiennent à la syntaxe abstraite étendue mais non à la syntaxe initiale) ne sont jamais envoyées à une version d'équipement qui ne prend en charge que le protocole initial.

Les modifications non compatibles en amont sont celles qui influent sur la syntaxe de transfert d'une manière non compatible. Dans ce cas, une PDU codée du protocole initial n'est pas nécessairement une PDU codée valable pour le nouveau protocole.

En général, une modification non compatible du point de vue de la syntaxe abstraite cause une modification non compatible du point de vue de la syntaxe de transfert. Cependant, pour un ensemble donné de règles de codage, il peut y avoir des exceptions.

Des exemples de ces modifications sont donnés au 12.5.1.3.

En outre, il est évident que la modification des règles de codage cause dans la plupart des cas une incompatibilité du point de vue de la syntaxe de transfert.

12.5.1.1 Modifications sans incidence sur la syntaxe abstraite

Ces modifications se limitent strictement à la façon dont la syntaxe abstraite et le type utilisé pour sa définition sont spécifiés. Elles n'influent pas sur l'ensemble de valeurs définies par la syntaxe abstraite. Ces modifications peuvent être nécessaires pour mettre une spécification en conformité avec les règles indiquées aux 12.5.1.2 et 12.5.1.3 du présent addendum (la liste de ces modifications énumérées ci-dessous n'est pas nécessairement exhaustive).

a) Dans un type d'ensemble ou un type de séquence, remplacer l'élément «COMPONENTS OF» en incluant directement des composants équivalents, ou vice versa.

b) Dans un type de choix, remplacer les types de choix emboîtés en incluant directement chaque élément «NamedType» qui figure dans la liste «AlternativeTypeList».

c) Remplacer un type par un élément «typereference» représentant le même type, ou vice versa.

d) Remplacer une valeur par un élément «valuereference» qui la représente, ou inversement, notamment, remplacer un élément «number» par un élément «valuereference».

e) Remplacer un type par un type de sélection équivalent et vice versa.

f) Ajouter (ou en cas de non-utilisation) supprimer un ou plusieurs éléments «NamedBit» dans un type de chaîne binaire.

g) Ajouter (ou en cas de non-utilisation) supprimer un ou plusieurs éléments «NamedNumber» dans un type de valeur entière.

h) Changer l'orthographe d'un élément «typereference», «modulereference», «valuereference» ou «identifier» systématiquement dans tous les modules ASN.1, notamment ajouter des identificateurs lorsque la syntaxe le permet.

i) Scinder un module ASN.1 en plusieurs modules ASN.1.

j) Rassembler plusieurs modules ASN.1 en un seul module ASN.1.

k) Transférer certaines parties d'un module ASN.1 dans un autre module ASN.1.

l) Ajouter un ou plusieurs éléments «Symbol» à la liste «EXPORTS» (ou supprimer la mention «EXPORTS» pour indiquer que tout est exporté).

m) Ajouter un ou plusieurs éléments «Symbol» à la liste «IMPORTS» (symboles provenant de modules déjà référencés dans la liste «IMPORTS» ainsi que des symboles provenant de modules ASN.1 nouvellement référencés).

n) Supprimer un ou plusieurs éléments «Valueassignment» existants si leur paramètre «valuereference» associé n'est jamais utilisé (même non implicitement dans une construction «ANY DEFINED BY») dans tous les modules ASN.1.

o) Supprimer un ou plusieurs éléments «Typeassignment» (y compris les éléments de type «OPERATION» et «ERROR») s'ils ne sont jamais utilisés dans tous les modules ASN.1.

p) Ajouter un type ou une valeur ERROR déjà incluse dans la syntaxe abstraite (c'est-à-dire associée à un autre type d'OPERATION) à la liste «ERRORS» d'un type «OPERATION» si ce type comportait une telle liste ou ajouter une liste «ERRORS» à un type «OPERATION» si ce type ne comportait pas une telle liste.

q) Ajouter un type ou une valeur OPERATION déjà incluse dans la syntaxe abstraite (par exemple, non pas comme une opération corrélée) à la liste «LINKED OPERATIONS» d'un type «OPERATION» si ce type comportait une telle liste ou ajouter une liste «LINKED OPERATIONS» à un type «OPERATION» si ce type ne comportait pas une telle liste.

12.5.1.2 Extension d'une syntaxe abstraite

Une syntaxe abstraite est étendue si son type associé est aussi étendu (par exemple, s'il s'agit d'un type de choix, celui-ci peut être étendu par l'adjonction d'un nouvel élément ou l'extension d'un élément existant). Pour étendre une PDU (ou n'importe quel type structuré), l'un des moyens consiste à étendre le type de l'un quelconque de ses éléments.

Un type d'ASN.1 est considéré comme l'extension d'un autre type si le premier inclut toutes les valeurs du second et éventuellement de certains autres.

Les extensions, en ce qui concerne un type donné, sont les types qui pourraient être obtenus par une ou plusieurs des modifications indiquées ci-après, combinées à un nombre quelconque de celles décrites au 12.5.1.1.

a) Convertir un type unique en un type de choix qui inclut ce type unique dans la liste «AlternativeTypeList».

NOTE 1 – L'étiquette de cette nouvelle version doit rester inchangée; aucune étiquette (EXPLICIT) complémentaire n'est admise. Il convient de veiller à ce que toutes les références à ce type modifié dans tous les modules ASN.1 répondent encore à toutes les conditions requises de l'ASN.1, en particulier la différenciation des étiquettes et l'unicité des identificateurs.

b) Ajouter un/plusieurs éléments «NamedType» à la liste «AlternativeTypeList» d'un type de choix.

NOTE 2 – Voir la Note relative à la conversion d'un type unique en un type de choix.

c) Ajouter un élément facultatif à un type de séquence ou à un type d'ensemble.

d) Ajouter un élément par défaut à un type de séquence ou à un type d'ensemble.

e) Etendre un ou plusieurs éléments d'un type de choix, d'un type de séquence ou d'un type d'ensemble.

f) Etendre le type en termes desquels un type «sequence-of» ou un type «set-of» est défini.

g) Convertir un élément obligatoire d'un type de séquence ou d'un type d'ensemble en un élément facultatif ou par défaut.

NOTE 3 –L'étiquette de cet élément doit rester inchangée et la différenciation des étiquettes doit être assurée.

h) Ajouter un ou plusieurs éléments «NamedNumber» nouveaux à un type énuméré.

NOTE 4 – La différenciation des valeurs et l'unicité des identificateurs doivent être assurées.

i) Etendre l'élément «ValueRange» d'un type de valeurs entières en diminuant le point «LowerEndpoint» et/ou en augmentant le point «UpperEndpoint».

j) Etendre l'élément «SizeConstraint» d'un type de chaîne d'octets, d'un type de chaîne d'éléments binaires ou d'une chaîne de caractères en diminuant le point «LowerEndpoint» et/ou en augmentant le point «UpperEndpoint».

k) Etendre l'élément «SizeConstraint» d'un type de séquence, d'un type d'ensemble en diminuant le point «LowerEndpoint» et/ou en augmentant le point «UpperEndpoint».

l) Modifier le paramètre «Value» assigné à un élément «Valuereference» si l'effet pour toutes les références reste conforme à toutes les autres règles (par exemple, augmenter un paramètre «Value» utilisé seulement comme point «UpperEndpoint» dans un élément «ValueRange» ou «SizeConstraint»).

Les modifications suivantes apportées à la définition des éléments «OPERATION» et «ERROR» influent sur la syntaxe abstraite formée par l'ensemble de valeurs dont le type est celui des messages TCAP ou des PDU ROSE paramétrés par une liste spécifique d'opérations.

m) Ajouter une nouvelle définition de valeur respectivement pour le type «OPERATION» et «ERROR» dès lors qu'elle est distincte de toutes les autres définitions de valeur respectivement pour le type «OPERATION» et «ERROR».

n) Ajouter un type «ARGUMENT/PARAMETER» à un type «OPERATION» si celui-ci ne comporte pas un tel élément «ARGUMENT/PARAMETER».

o) Ajouter un type «RESULT» à un type «OPERATION» si celui-ci comporte un élément «RESULT» vide ou ne comporte aucun élément «RESULT».

p) Ajouter un type «PARAMETER» à un type «ERROR» si celui-ci ne comporte aucun élément «PARAMETER».

12.5.1.3 Modifications non compatibles

Ces modifications causent une incompatibilité, du point de vue de la syntaxe abstraite, lorsqu'une valeur de la syntaxe abstraite initiale n'est pas une valeur valable pour la nouvelle syntaxe abstraite.

Une modification non compatible du ou des types en termes desquels une syntaxe abstraite est définie cause une incompatibilité entre la syntaxe abstraite initiale et la nouvelle syntaxe.

Quelques exemples de ces modifications non compatibles sont donnés dans la liste ci-dessous:

– remplacer un type par un autre type même si l'étiquette reste la même;
– supprimer une définition de type ou une définition de valeur indiquée explicitement (IMPORTS) ou implicitement («ANY DEFINED BY» du TCAP);
– supprimer un élément «NamedType» dans la liste «AlternativeTypeList» d'un type de choix;
– supprimer un ou plusieurs éléments «NamedNumber» dans l'«Enumeration» d'un type énuméré;
– restreindre l'élément «ValueRange» d'un type de valeurs entières;
– restreindre l'élément «SizeConstraint» d'un type de chaîne;
– restreindre l'élément «SizeConstraint» d'un type «sequence-of»;
– modifier l'ordre des éléments dans la liste «ElementTypeList» d'un type de séquence;
– apporter, sous la forme d'une quelconque combinaison, les modifications indiquées ci-dessus à un ou plusieurs éléments d'un type structuré.

12.5.2 Règles de compatibilité en aval

La compatibilité en aval est obtenue le plus souvent par la négociation du contexte d'application. Cependant, pour réduire au minimum le nombre de replis d'un protocole à l'autre dans le réseau de signalisation, il faudra parfois définir des règles de compatibilité en aval qui permettent à une version de protocole d'accepter des unités de données de protocole engendrées par une future version sans être obligé de fournir un nom de contexte d'application nouveau.

Cette caractéristique est également nécessaire lorsque la négociation du contexte d'application n'est pas assurée, par exemple lorsque la partie de dialogue facultative de la capacité de transaction (TC) n'est pas prise en charge.

Si un concepteur de protocoles désire s'assurer qu'une valeur d'une PDU de la nouvelle version d'un protocole sera (au moins) toujours partiellement reconnue par une ancienne version de ce protocole, il doit suivre les directives suivantes:

- la nouvelle version de protocole doit être conforme aux règles énoncées au 12.5.1 (c'est-à-dire que la nouvelle syntaxe abstraite doit être une extension de l'ancienne);
- les règles de codage applicables (qui doivent être inchangées) doivent permettre de délimiter les parties inconnues du codage[1)];
- des règles d'extensibilité doivent être incluses dans la spécification du protocole initial.

Si la dernière Recommandation n'est pas suivie, le comportement d'une entité réceptrice dépend de la mise en œuvre.

Il est recommandé de restreindre l'utilisation des règles d'extensibilité à:

- l'adjonction d'éléments «OPTIONAL» et «DEFAULT» dans les types dérivés du type «SEQUENCE» ou «SET»;
- l'adjonction de l'assignation d'éléments binaires dans les types dérivés du type «BIT STRING» (sans extension de la contrainte de longueur);
- l'adjonction de variantes d'un type «CHOICE», sous réserve que celui-ci ne corresponde pas à un élément obligatoire d'une structure plus élevée;
- l'adjonction de valeurs énumérées à un type «ENUMERATED», sous réserve que celui-ci ne corresponde pas à un élément obligatoire d'une structure plus élevée.

Un concepteur de protocoles doit toujours être conscient du fait que des règles d'extensibilité allant au-delà de celles qui sont énumérées ici (par exemple, assouplissement d'une contrainte de longueur ou d'une gamme de valeurs, ou extension d'un type «CHOICE» correspondant à un élément obligatoire) peuvent exiger des précautions particulières (par exemple, spécification de codes d'erreur à utiliser lorsqu'un élément d'information non reconnu est rencontré) pour éviter des erreurs fonctionnelles importantes.

---

1) Cette condition est toujours remplie en cas d'utilisation de BER.

Il existe essentiellement deux moyens de spécifier les règles d'extensibilité, à savoir:

i) La spécification comporte une déclaration générale qui s'applique globalement à toute PDU du protocole.

Cette façon de procéder est illustrée par le texte ci-dessous extrait de la Recommandation UIT-T X.862, traitement des transactions (TP).

«Pour assurer la compatibilité future, un TPPM récepteur doit:

a) ignorer tout élément indéfini d'une notation SET, ENUMERATED ou SEQUENCE;

b) lorsque des bits affectés d'un nom sont utilisés dans l'ASN.1, traiter tout élément binaire comme non significatif si aucun nom ne lui est attribué.»

ii) La spécification comporte plusieurs déclarations dont chacune s'applique particulièrement à une PDU donnée (ou à toute structure interne) du protocole.

La version 1993 de l'ASN.1 fournit une nouvelle notation pour signaler les types qui peuvent être étendus de telle sorte que les adjonctions inconnues soient ignorées. Ce fanion, appelé marqueur d'extension, est une ellipse (...). Il est recommandé aux utilisateurs de la version 1988 de l'ASN.1 d'inclure ce fanion à titre de commentaire dans l'ASN.1.

Les exemples suivants illustrent une spécification de 1993 où des valeurs d'élément supplémentaires (inconnues) seront acceptées pour les notations PDU-A et TypeA mais non pour la notation TypeB.

```
PDU-A ::= SEQUENCE {
        element1 TypeA,
        element2 TypeB,
        ...}

TypeA ::= SEQUENCE {
        element3 TypeC,
        element4 TypeD,
        ...
        }

TypeB ::= SEQUENCE {
        element5 TypeE,
        element6 TypeF }
```

Le mécanisme d'extensibilité peut être également nécessaire en cas d'intervention de relais de couche application qui prennent en charge une version de protocole inférieure à celle utilisée effectivement par les entités expéditrices et réceptrices des messages (par exemple, dialogue TC Vn-Vn retransmis par un noeud Vn-1). Cependant, dans ce cas, il faut en outre que le noeud-relais retransmette l'information non reconnue reçue au noeud suivant.

La spécification de règles d'extensibilité sans autre indication n'impose à l'entité réceptrice aucune contrainte en ce qui concerne le traitement ultérieur de la partie du codage qui correspond aux valeurs inconnues. En l'absence de toute spécification complémentaire, il est implicitement admis que les parties non décodées sont ignorées.

S'il est nécessaire de prendre d'autres dispositions particulières (par exemple, renvoi d'une erreur spécifique, retransmission de telle ou telle partie du codage à une tierce entité, ...), un concepteur de protocoles doit établir une spécification explicite du comportement prévu, en plus des règles d'extensibilité. A titre d'option, le concepteur de protocoles peut éventuellement inclure dans la version initiale du protocole certains éléments d'information donnant dynamiquement une indication du comportement à adopter en cas de réception d'éléments d'information non reconnus.