



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Z.120

Corrigendum 1
(12/2001)

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

Techniques de description formelle – Diagrammes des
séquences de messages

Diagrammes des séquences de messages

Corrigendum 1

Recommandation UIT-T Z.120 (1999) – 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 combinée arborescente et tabulaire	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.120

Diagrammes des séquences de messages

Corrigendum 1

Résumé

Voici la liste des modifications de la Rec. UIT-T Z.120 (MSC) qui ont été approuvées par la Commission d'études en septembre 2001.

Source

Le Corrigendum 1 de la Recommandation Z.120 de l'UIT-T, élaboré par la Commission d'études 10 (2001-2004) de l'UIT-T, a été approuvé le 14 décembre 2001 selon la procédure définie dans la Résolution 1 de l'AMNT.

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.

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 2002

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	Objectifs et domaine d'application	1
2	Terminologie	1
3	Maintenance de la Rec. UIT-T Z.120	1
4	Corrections de la Rec. UIT-T Z.120	2
4.1	Paragraphe 1.2: erreur matérielle – Référence à la Partie II.....	2
4.2	Paragraphe 1.3: clarification – Références à des productions non existantes dans les exemples de métalangage	2
4.3	Paragraphe 1.3: erreur matérielle – Terme erroné.....	2
4.4	Paragraphe 1.3: erreur matérielle – Guillemet fermant manquant	2
4.5	Paragraphe 1.4: clarification – Noms de productions génériques	2
4.6	Paragraphe 1.4: clarification – Production correcte d'exemple	2
4.7	Paragraphe 4.1: erreur matérielle – Référence à <event list> inexistant.....	2
4.8	Paragraphe 4.1: clarification – Suppression de la référence à l'interface entre instance et environnement	2
4.9	Paragraphe 6: clarification – Relation entre sémantiques temporisée et non temporisée.....	3
4.10	Paragraphe 2.1: modification – Règles lexicales.....	3
4.11	Paragraphe 2.1: modification – Suppression de mots clés	3
4.12	Paragraphe 2.1: modification – Remplacement des mots clés "startbefore", "startafter", "endbefore" et "endafter"	3
4.13	Paragraphe 2.1: erreur matérielle	3
4.14	Paragraphe 2.1: clarification – Explication relative au rôle des <space>.....	3
4.15	Paragraphe 2.1: clarification et modification – Production de <non par non escape>	4
4.16	Paragraphe 3: modification – Déplacement de la partie relative aux données dans le Document MSC.....	4
4.17	Paragraphe 3: clarification – Messages environnementaux dans des diagrammes MSC redéfinis	4
4.18	Paragraphe 4.1: modification – Déplacement de la référence <time interval> dans la <time dest list>	4
4.19	Paragraphe 4.1: modification – Ordre aléatoire des blocs de paramètres	4
4.20	Paragraphe 4.1, 4.2 et 2.1: modification et clarification – Suppression des mots clés <kind denominator> du SDL	5
4.21	Paragraphe 4.1: clarification – Remplacement de <textual msc document> par "MSC document"	5
4.22	Paragraphe 4.1: modification – Adjonction de <label names> à <time dest> et changement de mots clés	5

	Page
4.23 Paragraphe 4.2: clarification – Définition du moment à partir duquel des instances statiques commencent à exister.....	6
4.24 Paragraphe 4.2: erreur matérielle	6
4.25 Paragraphe 4.2: clarification – Instances de création et de terminaison à l'intérieur de la référence MSC.....	6
4.26 Paragraphe 4.2: erreur matérielle – Absence du titre Exigences statiques.....	6
4.27 Paragraphe 4.2: erreur matérielle	6
4.28 Paragraphe 4.3: erreur matérielle – Référence à des productions inexistantes	6
4.29 Paragraphe 4.4: erreur matérielle – Référence à une production non existante.....	7
4.30 Paragraphe 4.5: erreur matérielle	7
4.31 Paragraphe 8.1: erreur matérielle – Figures 23, 24, 25 et 26.....	7
4.32 Paragraphe 4.7: erreur matérielle	7
4.33 Paragraphe 4.7: clarification – Instances concernées par une condition	7
4.34 Paragraphe 4.8: clarification – Type d'expressions de durée de temporisation	7
4.35 Paragraphe 5.2: erreur matérielle – Remplacement des tirets par des espaces dans les règles de syntaxe.....	8
4.36 Paragraphe 5.2: modification – Adjonction de délimiteurs dans <escapechar>	8
4.37 Paragraphe 5.2: clarification – Remplacement du mot "delimiter"	8
4.38 Paragraphe 5.2: modification – Suppression de la virgule dans <par decl list>	8
4.39 Paragraphe 5.2: modification – Grammaire pour les déclarations de parenthèses	8
4.40 Paragraphe 5.2: clarification et erreur matérielle – Grammaire pour les déclarations de parenthèses	10
4.41 Paragraphe 7.3: modification – Changement de ";" terminateur en ";" séparateur.....	10
4.42 Paragraphe 5.5: modification – Mot clé "variables" facultatif dans <data parameter decl>	10
4.43 Paragraphe 6.4: modification – Adjonction de l'unité de temps par défaut....	10
4.44 Paragraphe 6.9: erreur matérielle – Erreurs de marques de temps	11
4.45 Paragraphe 6.10: erreur matérielle	11
4.46 Paragraphe 6.10: erreur matérielle – Références à des productions incorrectes.....	11
4.47 Paragraphe 6.10: erreur matérielle	11
4.48 Paragraphe 6.10: modification – Mot clé de int_boundary rendu facultatif...	11
4.49 Paragraphe 6.10: clarification – Signification des symboles de symétrie	11
4.50 Paragraphe 6.10: modification – Adjonction d'intervalles temporels absolus.....	11

	Page
4.51 Paragraphe 7.2: modification – Adjonction d'une séquence pour les expressions en ligne.....	12
4.52 Paragraphe 7.2: modification – Remplacement des mots clés de contrainte temporelle et adjonction d'une déclaration temporelle dans <shared inline expr>	13
4.53 Paragraphe 7.2: modification – Remplacement des mots clés de contrainte temporelle et adjonction d'une déclaration temporelle dans <inline expr>	13
4.54 Paragraphe 7.2: clarification – Explication relative aux contraintes temporelles dans les expressions en ligne	13
4.55 Paragraphe 7.3: modification – Remplacement des mots clés de contrainte temporelle et déplacement des déclarations temporelles dans <shared msc reference>	13
4.56 Paragraphe 7.3: modification – Adjonction d'une déclaration de contrainte temporelle et déplacement de la déclaration dans la règle <msc reference>	14
4.57 Paragraphe 7.3: clarification – Changement des mots clés de contrainte temporelle	14
4.58 Paragraphe 7.3: erreur matérielle – Absence de référence à "seq"	14
4.59 Paragraphe 7.3: erreur matérielle – Référence incorrecte à une production ..	14
4.60 Paragraphe 7.3: modification – Ordre aléatoire des blocs de paramètres réels.....	14
4.61 Paragraphe 7.4: erreur matérielle – Déclaration d'instance incorrecte dans les Figures 12 et 16.....	15
4.62 Paragraphe 7.5: modification – Adjonction de déclarations de contrainte temporelle dans les nœuds de référence HMSC et parallèles.....	15
4.63 Paragraphe 7.5: erreur matérielle – Référence incorrecte à une production ..	15
4.64 Section 11: erreur matérielle	15
4.65 Section 11: erreur matérielle – Suppression des unités de temps dans les Figures 43, 44 et 46	15
4.66 Section 11: erreur matérielle – Casse du mot clé msc dans les Figures 42 à 46	16
4.67 Section 11: erreur matérielle – Adjonction de texte manquant dans la Figure 42.....	16
4.68 Annexe A: erreur matérielle – Référence incorrecte à <keywords>	16
4.69 Point d'ordre général – Emploi de <end> ou du point virgule dans la grammaire textuelle	16

Recommandation UIT-T Z.120

Diagrammes des séquences de messages

Corrigendum 1

1 Objectifs et domaine d'application

Le présent corrigendum a pour objet de consigner les modifications de la Rec. UIT-T Z.120 qui ont été approuvées. La portée et la nature de ces modifications suivent le modèle de la classification présentée dans l'Appendice II/Z.100 (SDL).

Ces modifications sont de deux ordres:

- a) corrections d'*erreurs* et *clarifications*;
- b) *extensions* et *modifications*.

Les règles de maintenance énoncées dans l'Appendice II/Z.100 stipulent que les *corrections* et les *clarifications* publiées dans la liste principale des modifications "prendront effet immédiatement". De telles modifications doivent être publiées le plus rapidement possible dans un corrigendum.

Les *modifications* et les *extensions* dénotent des changements apportés aux diagrammes MSC. Dans ce cas, la règle est la suivante: "Sauf si des circonstances particulières nécessitent l'implémentation de tels changements le plus tôt possible, ceux-ci ne seront pas recommandés avant la prochaine révision de la Z.100 (sic)". Les *modifications* et les *extensions* dont il est question dans le présent corrigendum ne sont pas considérées comme des changements significatifs mais comme des améliorations utiles de la notation MSC; pour cette raison, il est recommandé de les appliquer sans attendre.

2 Terminologie

Une *erreur* (*error*) est une incohérence relevée dans la Rec. UIT-T Z.120.

Une *correction textuelle* (*textual correction*) est une modification du texte ou des diagrammes permettant de corriger des erreurs matérielles ou typographiques.

Un *sujet ouvert* (*open item*) est un problème identifié mais non résolu.

Une *défaillance* (*deficiency*) est un sujet identifié pour lequel la sémantique du SDL n'est pas (clairement) définie dans la Rec. UIT-T Z.100.

Une *clarification* (*clarification*) est une modification du texte (ou des diagrammes) de la Recommandation qui ne modifie pas (intentionnellement) le sens des MSC mais qui clarifie le texte ou les diagrammes qui pourraient être compris de manière ambiguë sans cette clarification.

Une *modification* (*modification*) est un changement apporté à la sémantique des MSC.

Une *extension* (*extension*) est une nouvelle caractéristique qui ne modifie pas la sémantique des caractéristiques définies dans la Recommandation approuvée.

3 Maintenance de la Rec. UIT-T Z.120

L'Appendice II/Z.100 (1999) expose la procédure à suivre pour la maintenance des Recommandations UIT-T de la série Z.100, utilisée comme base pour la maintenance de la Rec. UIT-T Z.120 (MSC). Conformément à cette procédure les corrections, propositions de modification et extensions doivent être largement diffusées et la liste principale des modifications doit être tenue

à jour. On trouvera dans le présent corrigendum une liste de modifications qui ne correspond pas toujours à la liste principale des modifications, laquelle contient également d'autres informations.

4 Corrections de la Rec. UIT-T Z.120

4.1 Paragraphe 1.2: erreur matérielle – Référence à la Partie II

"Partie II" changée en "Section 6".

4.2 Paragraphe 1.3: clarification – Références à des productions non existantes dans les exemples de métalangage

L'exemple de production `<msc document body>` du § 1.3 n'existe pas dans la grammaire; il est remplacé par la production `<using clause>`. Le texte explicatif accompagnant est changé en conséquence.

4.3 Paragraphe 1.3: erreur matérielle – Terme erroné

l'ordre d'importance est remplacé par *l'ordre de préséance*.

4.4 Paragraphe 1.3: erreur matérielle – Guillemet fermant manquant

Un guillemet fermant doit être ajouté à la dernière phrase du § 1.3: "{...}".

4.5 Paragraphe 1.4: clarification – Noms de productions génériques

Adjonction d'une phrase clarifiant le rôle des noms de productions génériques `<area>`, `<non terminal>`, etc.

4.6 Paragraphe 1.4: clarification – Production correcte d'exemple

La règle de production donnée en exemple pour `<message end area>` est corrigée afin de correspondre à la production indiquée dans la grammaire.

4.7 Paragraphe 4.1: erreur matérielle – Référence à `<event list>` inexistant

Le nom de la production `<event list>` est changé en `<instance event list>` dans la *Sémantique* du § 4.1.

4.8 Paragraphe 4.1: clarification – Suppression de la référence à l'interface entre instance et environnement

Supprimer les phrases:

"La déclaration facultative `<msc interface>` décrivant l'interface d'un diagramme MSC avec son environnement consiste en une interface de type `<msc inst interface>` et une interface de type `<msc gate interface>`."

et:

"Puisqu'un diagramme MSC ne représente que l'exécution d'une partie d'un système, l'interface `<msc inst interface>` décrit les points de connexion des instances à l'environnement."

Elles n'ont pas de sens et sont redondantes.

4.9 Paragraphe 6: clarification – Relation entre sémantiques temporisée et non temporisée

La dernière phrase du deuxième alinéa du § 6 devient

"On peut obtenir l'interprétation non temporisée d'un diagramme MSC à partir de l'interprétation temporisée en supprimant les événements temporels supplémentaires de ses traces; ainsi des groupes de traces temporisées différents se réduiront à la même trace non temporisée."

Ceci est une simple clarification.

4.10 Paragraphe 2.1: modification – Règles lexicales

Supprimer la production de <word> étant donné qu'elle n'est pas utilisée, même si <word> apparaît dans <lexical unit>, où il doit être remplacé par <name>, vu que celui-ci manque à cet endroit.

Déplacer la production de <qualifier> de la section 3 à 2.1, vu qu'il doit être traité de la même façon que <note>; par ailleurs, <text> n'est pas exporté en tant qu'unité lexicale et, pour cette raison, n'est pas visible dans le paragraphe relatif à l'analyse.

La référence à <qualifier> doit être ajoutée à la production de <lexical unit>.

La production de <misc> doit contenir entre autres <apostrophe>.

Déplacer "[", "]", "#", et "@" de <national> à <special> avec leurs noms de production lorsqu'ils sont définis.

Déplacer "&", "<", et ">" de <other characters> à <special> en utilisant leur nom de production.

4.11 Paragraphe 2.1: modification – Suppression de mots clés

Supprimer les mots clés **by** et **order** de la liste <keyword> étant donné qu'ils ne sont utilisés nulle part.

4.12 Paragraphe 2.1: modification – Remplacement des mots clés "startbefore", "startafter", "endbefore" et "endafter"

Les quatre mots clés **startbefore**, **startafter**, **endbefore** and **endafter** sont utilisés dans la Grammaire textuelle concrète pour indiquer l'endroit où il faut placer une contrainte de temporisation dans une expression en ligne. Toutefois, deux mots clés seulement sont nécessaires, et le sens de *before* ou de *after* ne peut être spécifié statiquement étant donné que les contraintes de temporisation s'appliquent dynamiquement. Autrement dit, on ne peut dire statiquement si une extrémité d'un intervalle est dynamiquement le premier ou le dernier événement par rapport à l'autre événement. Pour cette raison, les quatre mots clés ont été supprimés et remplacés par **top** et **bottom**, qui déterminent si la contrainte s'applique respectivement au premier ou au dernier événement de l'expression. Toutes les occurrences des anciens mots clés de la norme ont été mises à jour pour tenir compte de ce changement. Le changement de mot clé traduit le positionnement possible des contraintes de temporisation dans la grammaire graphique.

4.13 Paragraphe 2.1: erreur matérielle

Sans objet dans le texte français.

4.14 Paragraphe 2.1: clarification – Explication relative au rôle des <space>

La phrase suivante doit être ajoutée à la fin du deuxième alinéa du texte suivant la grammaire formelle du § 2.1, étant donné qu'il n'y a pas d'explication sur le rôle que jouent ces <space> dans la grammaire. Cette phrase fait pendant à une phrase de la norme SDL (Rec. UIT-T Z.100).

"Un nombre quelconque d'espaces <space> peut être inséré avant ou après toute unité <lexical unit>. Des <space> ou des <note> insérés n'ont aucune signification syntaxique mais il est parfois nécessaire d'utiliser un <space> ou une <note> pour séparer deux unités <lexical unit>."

4.15 Paragraphe 2.1: clarification et modification – Production de <non par non escape>

Le texte à droite de la règle de <non par non escape> est modifiée: a) pour éviter toute confusion entre l'utilisation informelle du terme "character string" étant donné qu'il s'agit du nom formel d'une production, b) pour éviter toute confusion du même ordre au sujet du mot "delimiter"; c) pour ajouter "J", qui manquait, à la liste des symboles de terminaison (il peut mettre fin à un intervalle temporel). Donc, "character string" devient simplement "string" et "parenthesis delimiters" devient "parenthesis string". Le côté droit de la production devient donc:

string containing no escape char and no parenthesis strings and not the possible terminators `"`, `'`, `\"`, `\"'` or **comment**

D'une manière analogue, la première occurrence du mot "delimiter" est changée en "string" dans la phrase qui précède immédiatement la production de <string>; la deuxième occurrence doit être supprimée. Le passage de "delimiter" à "string" doit également être fait dans des phrases précédant la production <non parenthesis> et enfin après la production <non par non escape>.

4.16 Paragraphe 3: modification – Déplacement de la partie relative aux données dans le Document MSC

Les productions de `<data definition>` et de `<parenthesis declaration>` doivent être déplacées du bas vers le haut de la production de `<document head>`. Le déplacement de `<parenthesis declaration>` facilite l'analyse lexicale du document et le déplacement de `<data definition>` facilite l'analyse statique de l'utilisation des données. Les diagrammes donnés en exemple sont modifiés pour faire apparaître ce changement grammatical. La référence `<data definition>` ne sera plus facultative étant donné que chacun des éléments de sa production est lui-même facultatif, ce qui supprime l'ambiguïté de la grammaire.

4.17 Paragraphe 3: clarification – Messages environnementaux dans des diagrammes MSC redéfinis

Dans les Exigences statiques du § 3, ajouter la phrase "Les messages environnementaux d'un diagramme MSC **redefined** doivent correspondre exactement à ceux du diagramme qui est redéfini; un nombre de messages plus élevé ou moins élevé vers ou depuis l'environnement n'est pas permis.

4.18 Paragraphe 4.1: modification – Déplacement de la référence <time interval> dans la <time dest list>

Dans la production de <time dest list>, déplacer la référence <time interval>, qui peut terminer une suite de données, de manière à la placer après la destination facultative <time dest>. Ainsi, la suite de données est terminée soit par une "," ou par un ";", qui est reconnaissable, plutôt que par un identificateur (le <event name>), dont le départ ne peut être reconnu. La production se présente dès lors de la manière suivante:

$$\langle \text{time dest list} \rangle ::= [\langle \text{time dest} \rangle] \langle \text{time interval} \rangle [, \langle \text{time dest list} \rangle]$$

4.19 Paragraphe 4.1: modification – Ordre aléatoire des blocs de paramètres

Actuellement la grammaire fixe l'ordre des déclarations des paramètres "variable", "timer", "message" et "instance". La grammaire doit être changée pour permettre de déclarer ces blocs dans n'importe quel ordre. Concrètement, il faut changer la production `<msc parameter decl>` et ajouter les nouvelles productions intermédiaires `<msc parm decl list>` et `<msc parm decl block>`. Les productions doivent être:

```

<msc parameter decl> ::=
    ( <msc parm decl list> )
<msc parm decl list> ::=
    <msc parm decl block> [ <end> <msc parm decl list> ]
<msc parm decl block>h ::=
    <data parameter decl>
    | <instance parameter decl>
    | <message parameter decl>
    | <timer parameter decl>

```

La phrase suivante doit être ajoutée aux Exigences statiques:

"Les blocs de déclarations de paramètres de diagramme MSC figurant dans une liste <msc parm decl list>, à savoir variable, instance, message et timer peuvent être donnés dans n'importe quel ordre mais ils ne peut y avoir qu'un seul bloc par type. Par exemple, on ne peut avoir deux blocs paramètres de message même s'ils sont séparés par un bloc de déclarations d'instance".

Un changement correspondant pour les paramètres effectifs doit être fait dans le 7.3; ce point est traité dans la partie appropriée du présent corrigendum.

4.20 Paragraphes 4.1, 4.2 et 2.1: modification et clarification – Suppression des mots clés <kind denominator> du SDL

Les références explicites aux entités SDL doivent être supprimées étant donné qu'elles peuvent changer, comme c'est le cas de la suppression des services SDL. La grammaire MSC contient un mécanisme qui permet à un utilisateur d'écrire tout identificateur à la place des mots clés SDL spécifiques, de telle manière que la connexion puisse encore être faite de manière informelle. Les mots clés **system**, **block**, **process** et **service** doivent être enlevés de la production de <kind denominator> 4.1 et de la liste des mots clés dans la production des <keywords> du 2.1.

Dans la partie introductive 4.2, supprimer la phrase "Si on se réfère au SDL, une entité peut être un processus SDL, un bloc, un processus ou un service".

L'explication relative aux rôles de l'identificateur <instance kind> et de <kind denominator> a été améliorée dans la Sémantique de la section 4.2. Le deuxième alinéa a été scindé en deux, la deuxième partie est inchangée mais la première devient:

"Dans le cadre du SDL, une instance peut se rapporter à des entités telles que des processus, des blocs et des systèmes, et hors de ce cadre, elle peut se rapporter à tout type d'entité. Le <kind denominator> permet à l'utilisateur de nommer le type d'entité auquel se rapporte l'instance, par exemple un processus, un bloc, etc., et l'identificateur <instance kind> peut être utilisé pour donner le nom particulier de cette classe d'entité. Plusieurs instances peuvent partager le même identificateur <instance kind> pour dénoter qu'elles ont une nature commune."

4.21 Paragraphe 4.1: clarification – Remplacement de <textual msc document> par "MSC document"

La référence à <textual msc document> dans les Exigences statiques de la grammaire textuelle est remplacée par "MSC document". Comme la grammaire textuelle est censée être obtenue à partir de la représentation graphique, il est faux de vouloir qu'un élément provienne de la grammaire textuelle alors que l'équivalent graphique existe.

4.22 Paragraphe 4.1: modification – Adjonction de <label names> à <time dest> et changement de mots clés

La production de <time dest> dans la Grammaire textuelle concrète ne permet actuellement pas aux contraintes temporelles de se référer à des nœuds HMSC, ce qui n'est pas le but. Pour cette raison, il faut ajouter <label name> à la règle. De plus, les mots clés **begin** et **end** doivent être remplacés par

top et **bottom** par souci d'uniformité avec d'autres règles et leur caractère facultatif doit être supprimé, ce qui obligera à donner un mot clé au moins. La règle pour <time dest> devient:

<time dest> ::=
 <event name> | { **top** | **bottom** } { <reference identification> | <label name> }

Une exigence statique additionnelle doit être ajoutée par suite du changement apporté à cette règle afin de s'assurer que le nœud de destination d'une <time dest> dans un HMSC est un <timeable node> et non un <node>. La phrase suivante est ajoutée à la fin des Exigences statiques qui suit la Grammaire textuelle concrète:

"Les noms <label name> utilisés dans une <time dest> doivent se référer uniquement à des <timeable node>."

4.23 Paragraphe 4.2: clarification – Définition du moment à partir duquel des instances statiques commencent à exister

La phrase suivante est ajoutée à la fin du premier alinéa de la partie Sémantique pour clarifier le moment auquel les instances ordinaires, créées de manière non dynamique, deviennent existantes.

"Les instances statiques d'un document MSC sont les instances qui apparaissent dans sa partie définissante mais qui ne sont pas créées dynamiquement. Les instances statiques sont créées au moment de la création de leur instance englobante (sorte), celle-ci étant le nom du document MSC."

4.24 Paragraphe 4.2: erreur matérielle

Sans objet dans le texte français.

4.25 Paragraphe 4.2: clarification – Instances de création et de terminaison à l'intérieur de la référence MSC

Remplacer la dernière phrase précédant la partie Sémantique par:

"Si une instance est créée mais qu'elle n'est pas terminée à l'intérieur d'un diagramme MSC référencé, l'en-tête et l'axe d'instance doivent être inclus dans les diagrammes MSC référants et aucun événement ne peut être placé entre l'en-tête et la référence. Cela permet de continuer l'axe d'instance au-dessous de la référence pour être identifié avec le nom d'instance correct. Inversement, si une instance est terminée (arrêtée) mais non créée dans un MSC référencé, l'axe d'instance doit être terminé au niveau du MSC référant et aucun événement ne peut être placé entre la référence et le symbole d'arrêt. Ces règles font qu'une référence a toujours le même nombre d'axes d'instance joints en dessous et au-dessus."

Cette explication élargie clarifie la manière dont sont représentées les instances qui, à défaut, n'émergeraient que d'un côté d'un symbole de référence MSC.

4.26 Paragraphe 4.2: erreur matérielle – Absence du titre Exigences statiques

Le texte qui suit immédiatement la Grammaire graphique concrète et qui commence par "L'en-tête d'instance <instance heading> ..." doit être précédé du titre Exigences statiques.

4.27 Paragraphe 4.2: erreur matérielle

Sans objet dans le texte français.

4.28 Paragraphe 4.3: erreur matérielle – Référence à des productions inexistantes

Changer la référence de la production de <name> en <gate name> dans le texte qui accompagne la Grammaire textuelle concrète. Changer la référence <address> en "<input address> or <output address>" dans la Grammaire textuelle concrète. Changer la référence <bindings> en "<binding>s"

dans Exigences statiques. Changer la référence <output event area> en "output <event area>" dans les Exigences statiques. De manière analogue, <input event area> devient "input <event area>".

4.29 Paragraphe 4.4: erreur matérielle – Référence à une production non existante

Changer la référence de la production de <call out event> en "<call out> event" dans les Exigences statiques.

4.30 Paragraphe 4.5: erreur matérielle

Sans objet dans le texte français.

4.31 Paragraphe 8.1: erreur matérielle – Figures 23, 24, 25 et 26

Les guillemets des noms de messages *PleaseEnter* et *TryAgain* doivent être supprimés dans les Figures 23, 24, 25 et 26. Les noms de message *Cardid*, *try_again*, et "*Please enter*" de la Figure 24 (et le dernier aussi dans la Figure 23) doivent être changés afin de lire *CardId*, *TryAgain* et *PleaseEnter*, respectivement.

4.32 Paragraphe 4.7: erreur matérielle

Sans objet dans le texte français.

4.33 Paragraphe 4.7: clarification – Instances concernées par une condition

La relation entre les instances concernées par un symbole de condition graphique et celles définies dans la liste <shared> qu'il englobe a été clarifiée.

La phrase suivante qui figure dans la Grammaire graphique concrète:

"Les instances partagées <shared> ne possèdent pas de zone d'instance <instance area> dans le diagramme car il n'y a pas d'événements relatifs à ces instances mais ces instances partagées restent couvertes par la condition."

est remplacée par:

"Les instances couvertes par une condition sont celles qui lui sont rattachées dans le diagramme et celles énumérées explicitement dans la clause <shared>."

La dernière phrase est clarifiée et déplacée dans les Exigences statiques, où elle devient:

"S'il y a une quelconque ambiguïté concernant l'ordre des événements des instances <shared> dues à un symbole de condition implicite, elles doivent être représentées explicitement dans le diagramme avec une <instance area> plutôt que par le biais d'une liste <shared list>".

La phrase suivante doit être ajoutée aux Exigences statiques qui font suite à la Grammaire graphique concrète:

"Une instance qui est graphiquement couverte par un symbole de condition ne peut pas être incluse par la voie de sa liste <shared>".

4.34 Paragraphe 4.8: clarification – Type d'expressions de durée de temporisation

La phrase ci-après doit être ajoutée aux Exigences statiques. Elle précise que le type d'expression utilisée pour définir la durée d'un temporisateur est du type supposé "Time".

"Les suites <expression string> utilisées pour spécifier une <durée> doivent avoir le type supposé Time."

4.35 Paragraphe 5.2: erreur matérielle – Remplacement des tirets par des espaces dans les règles de syntaxe

Toutes les occurrences des noms de production suivantes sont changées par suppression du tiret:

<non-parenthesis>	devient	<non parenthesis>.
<non-nestable par>	devient	<non nestable par>.
<non-parenthesis>	devient	<non parenthesis>.
<non-par-non-escape>	devient	<non par non escape>.
<non-orderable event>	devient	<non orderable event>.
<non-nestable par pair>	devient	<non nestable par pair>.
<extra-global>	devient	<extra global>.

4.36 Paragraphe 5.2: modification – Adjonction de délimiteurs dans <escapechar>

La production de <escapechar> est changée de <character string> en <character string> entre deux délimiteurs <delim>. Cela pour deux raisons: a) une chaîne de caractères peut contenir des espaces, ce qui rend difficile de déterminer la fin de la chaîne, surtout si un espace est inclus dans la chaîne d'échappement; il n'y a pas de règles statiques limitant le nombre de caractères ou leur établissement dans la chaîne, et b) l'absence d'unité avec d'autres déclarations dans le bloc de paramètres.

La nouvelle production devient:

<escapechar> ::= <delim> <character string> <delim>

4.37 Paragraphe 5.2: clarification – Remplacement du mot "delimiter"

Dans la Sémantique du 5.2, remplacer "le délimiteur droit correspondant" par "les caractères entre parenthèses de droite correspondants" dans la phrase "L'analyseur lexical recherchera le délimiteur droit correspondant". Ce changement est nécessaire parce que, dans ce paragraphe, *délimiteur* a un sens formel en tant que caractères qui entourent les déclarations des chaînes de parenthèses, et que la référence s'applique aux caractères entre parenthèses et non aux caractères délimiteurs.

4.38 Paragraphe 5.2: modification – Suppression de la virgule dans <par decl list>

La production <par decl list> doit être changée en ce sens qu'il faut supprimer la virgule superflue qui sépare les déclarations qui constituent la liste <par decl list> étant donné que chaque déclaration est de toute manière terminée par un point virgule. Donc, dans la règle <par decl list>, la partie finale est changée de:

[, <par decl list>]

en

[<par decl list>].

4.39 Paragraphe 5.2: modification – Grammaire pour les déclarations de parenthèses

La grammaire pour déclarer les caractères de parenthèses est modifiée étant donné qu'elle ne rend pas l'effet recherché. En plus des virgules superflues supprimées selon la modification du 4.38, chaque paire appariée de caractères de parenthèses doit être annoncée dans une déclaration séparée. L'intention était que toutes les paires de parenthèses imbriquables, etc., devaient être annoncées dans cette seule déclaration.

Pour en tenir compte, la syntaxe pour déclarer les changements de paires de crochets est modifiée par suppression de la virgule qui sépare les parenthèses d'ouverture et de fermeture. A cet effet, le caractère délimiteur devient le séparateur et on obtient ainsi "()" à la place de "(,)".

L'exemple suivant concorde avec la syntaxe publiée:

```

parenthesis
  nestable      '(' , ')' ; ,
  nestable      '[' , ']' ; ,
  nonnestable   '/' , '*' / ; ,
  equalpar      '[' ; ,
  escape        \ ;
;

```

Dans la grammaire modifiée, cet exemple devient:

```

parenthesis
  nestable      '()' , '[' ] ;
  nonnestable   '/' '*' / ;
  equalpar      '[' ;
  escape        '\ ' ;

```

Les productions qui constituent la Grammaire textuelle concrète du 5.2 sont remplacées par:

```

<parenthesis declaration> ::=
    parenthesis <par decl list> <end>

<par decl list> ::=
    { <nestable par pair> | <non nestable par pair> | <equal par decl> | <escape decl> }
    [<par decl list>]

<nestable par pair> ::=
    nestable <pair par list> <end>

<non nestable par pair> ::=
    nonnestable <pair par list> <end>

<equal par decl> ::=
    equalpar <equal par list> <end>

<escape decl> ::=
    escape <escapechar>

<pair par list> ::=
    <pair par> [ , <pair par list> ]

<pair par> ::=
    <delim> <open par> <delim> <close par> <delim>

<equal par list> ::=
    <equal par> [ , <equal par list> ]

<equal par> ::=
    <delim> <par> <delim>

<delim> ::=
    | <apostrophe>
    | <alphanumeric>
    | <other character>
    | <special>
    | <full stop>
    | <underline>

<par> ::=
    <character string>

<escapechar> ::=
    <delim> <character string> <delim>

```

4.40 Paragraphe 5.2: clarification et erreur matérielle – Grammaire pour les déclarations de parenthèses

Le titre Exigences statiques manquant est introduit après les productions de la Grammaire textuelle concrète. La phrase expliquant les exigences statiques est remplacée par une autre, plus complète et plus claire, qui se présente ainsi:

"Le caractère délimiteur doit être le même caractère à gauche, au centre et à droite dans une production <pair par> et, d'une manière analogue, à gauche et à droite d'une production <equal par> ou <escapechar>, mais il peut être différent d'une occurrence à l'autre de ces productions. Les caractères délimiteurs ne doivent pas être contenus dans les chaînes de caractères qu'ils englobent. Ce qui suit, par exemple, est légal:

nestable '(') , /[/] ;

Dans l'analyse des déclarations de caractères de parenthèse, un analyseur MSC lit d'abord le premier caractère délimiteur et ensuite les autres caractères jusqu'à, sans l'inclure, l'occurrence suivante de ce délimiteur et considère cela comme la chaîne de parenthèses. Dans le cas d'une parenthèse appariée, l'analyseur considérera tous les caractères suivant le délimiteur central jusqu'à, sans l'inclure, la troisième occurrence du délimiteur comme chaîne de parenthèses de fermeture appariée. Aucun mécanisme d'échappement n'est prévu pour les chaînes de caractères entre les délimiteurs étant qu'un caractère délimiteur qui n'est pas contenu dans les chaînes de parenthèses doit être sélectionné par l'utilisateur pour éviter tout conflit."

4.41 Paragraphe 7.3: modification – Changement de ";" terminateur en ";" séparateur

Le <end> des productions <actual instance parameters>, <actual message parameters> et <actual timer parameters> a été supprimé parce que <actual parameters> a changé afin de permettre aux blocs de déclarations de se produire dans n'importe quel ordre.

4.42 Paragraphe 5.5: modification – Mot clé "variables" facultatif dans <data parameter decl>

Le changement en <msc parm decl> signifie que <data parameter decl> ne doit pas être le premier bloc de déclaration; il faut donc ajouter un mot clé facultatif initial **variables** à sa production. Cela s'applique également à <actual data parameters>. Le mot clé indiquera qu'un bloc de déclarations précédent est terminé car, à défaut, la grammaire serait ambiguë. L'adjonction du mot clé doit être faite dans la Grammaire textuelle concrète et les productions deviennent donc:

<data parameter decl> ::=
[**variables**] <variable decl list>

<actual data parameters> ::=
[**variables**] <actual data parameter list>

La phrase suivante doit être ajoutée après la première phrase des Exigences statiques:

"Si les déclarations de variables suivent un bloc de déclarations de temporisateur ou de message, le mot clé **variables** est utilisé pour indiquer que le bloc précédent est complet. Si les déclarations de variables se produisent en premier parmi les déclarations de paramètres, le mot clé **variables** est facultatif."

La phrase suivante doit être ajoutée à la fin des Exigences statiques:

"L'utilisation du mot clé **variables** dans <actual data parameter list> suit les règles s'appliquant à <data parameter decl>."

4.43 Paragraphe 6.4: modification – Adjonction de l'unité de temps par défaut

Aucune unité par défaut n'a été définie pour le temps. La phrase suivante, qui doit être ajoutée à la fin du § 6.4, fixe cette unité de temps par défaut comme étant la seconde:

"Les unités du domaine temporel et la représentation d'éléments de celui-ci peuvent être spécifiées via l'interface de données MSC (voir le § 5.4). Toutefois, si aucune unité de temps n'est définie à cet endroit, l'unité de temps par défaut est la seconde."

4.44 Paragraphe 6.9: erreur matérielle – Erreurs de marques de temps

Dans le texte introductif du § 6.9, les marques de temps absolu et relatif sont incorrectes. Le signe '&' doit être changé en '@' et le signe '?' en '&'.

4.45 Paragraphe 6.10: erreur matérielle

Sans objet dans le texte français.

4.46 Paragraphe 6.10: erreur matérielle – Références à des productions incorrectes

Remplacer la référence `<int_symbol 1>` par `<int symbol 1>` dans le texte accompagnant la Grammaire graphique concrète. Remplacer également `<int_symbol 2>` par `<int symbol 2>`.

4.47 Paragraphe 6.10: erreur matérielle

Sans objet dans le texte français.

4.48 Paragraphe 6.10: modification – Mot clé de `int_boundary` rendu facultatif

Le mot clé `int_boundary` est uniquement nécessaire dans la syntaxe ASCII, raison pour laquelle il doit être rendu facultatif dans la production de `<interval label>` du 6.10. La phrase suivante doit être ajoutée aux Exigences statiques pour expliquer ce qui précède:

"Lorsqu'une étiquette d'intervalle est utilisée, le mot clé `int_boundary` doit apparaître dans la représentation de programmation et doit être absent de la représentation graphique."

4.49 Paragraphe 6.10: clarification – Signification des symboles de symétrie

Dans la Grammaire graphique concrète, après la règle de production de `<int symbol 2>`, la partie de phrase "peuvent être projetés en symétrie horizontalement et verticalement" doit être remplacée par "peuvent être projetées en symétrie par rapport à un axe horizontal ou vertical". D'une manière analogue, dans le texte qui suit la définition de la production de `<cont int symbol>`, la partie de phrase "peut être projeté en symétrie verticalement" doit être remplacée par "peut être projeté en symétrie par rapport à un axe horizontal". Ces changements clarifient la signification de la projection en symétrie.

4.50 Paragraphe 6.10: modification – Adjonction d'intervalles temporels absolus

La grammaire graphique actuelle ne permet pas les intervalles temporels absolus, contrairement à la grammaire textuelle. Pour cette raison, la grammaire graphique doit être complétée par l'utilisation d'intervalles temporels absolus pour les contraintes temporelles. Deux nouvelles productions, `<abs time interval>` et `<abs bounded time>` doivent être ajoutées. Le second définit la syntaxe d'un intervalle temporel absolu, le premier remplace les choix explicites dans la production `<abs time are>`. Autrement dit, la partie `{ <abs time expression> | <abs measurement> }` de la règle `<abs time area>` est remplacée par `<abs time interval>`, qui complète le choix précédent avec `<abs bounded time>`. La nouvelle grammaire est donnée par:

```

<abs time area>::=
    <abs time symbol>
    is associated with <abs time interval>
    is attached to
    { <message out symbol> | <message in symbol> | <action symbol> |
      <timer start symbol> | <timer stop symbol> | <timeout symbol> |
      <inline expression symbol> | <separator symbol> |
      <msc reference symbol> | <par frame symbol> |
      <call in symbol> | <call out symbol> | <reply symbol> }

<abs time interval>::=
    <abs time expr> | <abs bounded time> | <abs measurement>

<abs bounded time>::=
    <abs time mark> { <left open> | <left closed> }
    [ <time point> ] , [ <time point> ]
    { <right open> | <right closed> }

```

4.51 Paragraphe 7.2: modification – Adjunction d'une séquence pour les expressions en ligne

Aucune séquence n'a été introduite dans la grammaire des expressions en ligne, bien que le texte s'y réfère. Etant donné que les séquences peuvent être utilement employées pour délimiter une région ou des événements afin d'appliquer des contraintes temporelles, elles doivent être ajoutées à la grammaire.

Dans la phrase introductive du § 7.2, "composition en parallèle" devient "composition en parallèle et séquentielle". Dans la phrase introductive de la Sémantique, "la composition en alternative, la composition en parallèle" devient "la composition en alternative, la composition séquentielle, la composition en parallèle". Dans le dernier alinéa de la Sémantique, "**alt end** et **par end**" devient "**alt end**, **seq end** et **par end**".

Les deux productions suivantes ont été ajoutées à la Grammaire textuelle concrète.

```

<shared seq expr> ::=
    seq begin [ <inline expr identification> ] <shared> <end>
    [ <inline gate interface> ] [ <instance event list> ]
    { seq <end> [ <inline gate interface> ] [ <instance event list> ] } *
    seq end <end>

<seq expr> ::=
    seq begin [ <inline expr identification> ] <end>
    [ <inline gate interface> ] <msc body>
    { seq <end> [ <inline gate interface> ] <msc body> } *
    seq end <end>

```

De plus, une référence à <shared seq expr> est ajoutée entre <shared alt expr> et <shared par expr> dans la règle de production de <shared inline expr>. D'une manière analogue, une référence à <seq expr> est ajoutée entre <alt expr> et <par expr> dans la règle de production de <inline expr>.

La production suivante est ajoutée à la Grammaire graphique concrète du 7.2.

```

<seq area> ::=
    <inline expression symbol> [is attached to <time interval area>] contains
    { seq <operand area>
      { is followed by <separator area> is followed by <operand area> } * }
    is attached to { <instance axis symbol> * } set
    is attached to { <inline gate area> * | <inline order gate area> * } set

```

Par ailleurs, une référence à <seq area> a été ajoutée à la règle de production de <inline expression area> entre <opt area> et <par area>.

4.52 Paragraphe 7.2: modification – Remplacement des mots clés de contrainte temporelle et adjonction d'une déclaration temporelle dans <shared inline expr>

La règle <shared inline expr> de la Grammaire textuelle concrète du 7.2 remplace les quatre déclarations liées aux mots clés **startbefore**, **startafter**, **endbefore** et **endafter**, par deux déclarations utilisant les mots clés **top** et **bottom**. Une déclaration d'intervalle temporel a été ajoutée; elle permet de supprimer les occurrences de <time interval> de chacune des productions suivantes pour les expressions partagées: <shared loop expr>, <shared opt expr>, <shared exc epr>, <shared alt expr> et <shared par expr> (Noter que: <shared seq expr>, étant une nouvelle règle, a tenu compte de ce changement). La nouvelle règle pour <shared inline expr> devient:

```
<shared inline expr> ::=
    [<extra global>]{ <shared loop expr> | <shared opt expr> |
    <shared alt expr> | <shared seq expr> | <shared par expr> | <shared exc expr> }
    [ time <time interval> <end>]
    [ top <time dest list> <end>]
    [ bottom <time dest list> <end>]
```

4.53 Paragraphe 7.2: modification – Remplacement des mots clés de contrainte temporelle et adjonction d'une déclaration temporelle dans <inline expr>

La règle <inline expr> de la Grammaire textuelle concrète du 7.2 remplace les quatre déclarations liées aux mots clés **startbefore**, **startafter**, **endbefore** et **endafter**, par deux déclarations utilisant les mots clés **top** et **bottom**. Une déclaration d'intervalle temporel a été ajoutée; elle permet de supprimer les occurrences de <time interval> de chacune des productions suivantes pour les expressions partagées: <loop expr>, <opt expr>, <exc epr>, <alt expt>, <alt expr> et <par expr> (Noter que: <seq expr>, étant une nouvelle règle, a tenu compte de ce changement). La nouvelle règle pour <inline expr> devient:

```
<inline expr> ::=
    [<extra global>]{ <loop expr> | <opt expr> | <alt expr> |
    <seq expr> | < par expr> | <exc expr> }
    [ time <time interval> <end>]
    [ top <time dest list> <end>]
    [ bottom <time dest list> <end>]
```

4.54 Paragraphe 7.2: clarification – Explication relative aux contraintes temporelles dans les expressions en ligne

L'avant-dernière phrase de la partie Sémantique du 7.2 est remplacée pour tenir compte de la grammaire modifiée dans laquelle les mots clés **startbefore**, **startafter**, **endbefore** et **endafter** ont été réduits à **top** et **bottom**, et aussi pour clarifier la relation entre la contrainte temporelle et les événements auxquels s'applique la contrainte.

La phrase "Dans la syntaxe textuelle, les mots clés ... selon le cas" est remplacée par:

"Dans la syntaxe textuelle, les mots clés **top** et **bottom** sont utilisés pour localiser une contrainte temporelle. Une contrainte temporelle liée au sommet d'une expression en ligne se réfère dynamiquement au premier événement qui survient dans cette expression, et inversement une contrainte temporelle liée au bas d'une expression en ligne se réfère dynamiquement au dernier événement qui survient à l'intérieur de l'expression en ligne."

4.55 Paragraphe 7.3: modification – Remplacement des mots clés de contrainte temporelle et déplacement des déclarations temporelles dans <shared msc reference>

La règle <shared msc reference> de la Grammaire textuelle concrète du 7.3 remplace les quatre déclarations liées aux mots clés **startbefore**, **startafter**, **endbefore** et **endafter** par deux déclarations utilisant les mots clés **top** et **bottom**. La déclaration d'intervalle temporel est déplacée à

l'intérieur de la règle pour rendre la grammaire analogue à celle des expressions en ligne. La nouvelle règle devient:

```
<shared msc reference> ::=
    reference [ <msc reference identification> : ]
    <msc ref expr> <shared> <end>
    [ time <time interval> <end> ]
    [ top <time dest list> <end> ]
    [ bottom <time dest list> <end> ]
    <reference gate interface>
```

4.56 Paragraphe 7.3: modification – Adjonction d'une déclaration de contrainte temporelle et déplacement de la déclaration dans la règle <msc reference>

La règle <msc reference> de la Grammaire textuelle concrète du 7.3 ne contient pas de déclaration de contrainte temporelle; pour cette raison, il convient d'ajouter deux déclarations utilisant les mots clés **top** et **bottom**. La déclaration d'intervalle temporel est déplacée dans la règle pour rendre la grammaire analogue à celle des expressions en ligne et des références msc partagées. La nouvelle règle devient:

```
<msc reference> ::=
    reference [ <msc reference identification> : ]
    <msc ref expr> <end>
    [ time <time interval> <end> ]
    [ top <time dest list> <end> ]
    [ bottom <time dest list> <end> ]
    <reference gate interface>
```

4.57 Paragraphe 7.3: clarification – Changement des mots clés de contrainte temporelle

La dernière phrase de la Sémantique du 7.3 doit être modifiée pour refléter les mots clés qui ont changé dans la grammaire. La liste des mots clés "**startbefore**, **startafter**, **endbefore** et **endafter**" est remplacée par "**top** et **bottom**".

4.58 Paragraphe 7.3: erreur matérielle – Absence de référence à "seq"

Dans la partie introductive de la section 7.3, l'opérateur "**seq**" manque dans la phrase "Les opérateurs **alt**, **par**, **loop**, **opt** et **exc** sont décrits en 7.2." Il convient donc d'ajouter l'opérateur "**seq**" à la phrase, qui devient "Les opérateurs **alt**, **par**, **seq**, **loop**, **opt** et **exc** sont décrits en 7.2".

4.59 Paragraphe 7.3: erreur matérielle – Référence incorrecte à une production

Remplacer la référence à la production <parameter declaration>, inexistante, par <msc parameter decl> dans les Exigences statiques du 7.3.

4.60 Paragraphe 7.3: modification – Ordre aléatoire des blocs de paramètres réels

L'ordre dans lequel les blocs de paramètres réels variable, timer, message et instance sont fournis est actuellement fixé par la grammaire. Celle-ci doit être modifiée afin de permettre à ces blocs d'être déclarés dans n'importe quel ordre. Concrètement, il y a lieu de modifier la production <actual parameters> et d'introduire les nouvelles productions intermédiaires <actual parameters list> et <actual parameters block>. Les productions seront:

```
<actual parameters> ::=
    ( <actual parameters list> )

<actual parameters list> ::=
    <actual parameters block> [ <end> <actual parameters list> ]
```

```

<actual parameters block> ::=
    <actual data parameters>
    | <actual instance parameters>
    | <actual message parameters>
    | <actual timer parameters>

```

Dans les Exigences statiques, il y a lieu de modifier la phrase suivante comme indiqué et d'ajouter la deuxième phrase, soit:

"Les éléments séparés par le symbole <end> dans les <actual parameters> doivent correspondre de façon biunivoque aux éléments séparés par le symbole <end> dans la déclaration <msc parameter decl> de la définition MSC, mais l'ordre des blocs peut être différent. Dans un bloc de paramètres, l'ordre des paramètres individuels doit être identique pour la déclaration et pour l'utilisation."

4.61 Paragraphe 7.4: erreur matérielle – Déclaration d'instance incorrecte dans les Figures 12 et 16

Les déclarations d'instance dans les documents MSC illustrées dans les Figures 12 et 16 sont incorrectes. La forme "**containing** i, j;" doit être remplacée par "**inst** i; **inst** j;". Les déclarations de message contiennent également le mot clé erroné, mais elles doivent être supprimées plutôt que corrigées étant donné que les déclarations de message sont facultatives pour les messages non paramétrés.

4.62 Paragraphe 7.5: modification – Adjunction de déclarations de contrainte temporelle dans les nœuds de référence HMSC et parallèles

La règle <node> de la Grammaire textuelle concrète du 7.5 ne contient pas les déclarations de contrainte temporelle s'appliquant aux nœuds de référence MSC et aux nœuds parallèles. Pour cette raison, il convient de modifier la règle <node expression> afin de distinguer les nœuds <node> non temporels des nouveaux <timeable node>. La règle <node> modifiée supprime les expressions de référence et les nœuds parallèles qui sont placés dans la nouvelle règle <timeable node>. Ces derniers ont des déclarations de contrainte temporelle exprimées de la même manière que pour les constructions MSC de base équivalente. Les nouvelles règles sont:

```

<node expression> ::=
    <label name> : { {<timeable node> | <node> } seq ( <label name list> )
    | end }<end>

<timeable node> ::=
    { ( <msc ref expr> ) | <par expression> }
    [ time <time interval> <end> ]
    [ top <time dest list> <end> ]
    [ bottom <time dest list> <end> ]

<node> ::=
    <condition identification>
    | connect

```

4.63 Paragraphe 7.5: erreur matérielle – Référence incorrecte à une production

Remplacer les références <label name> par <label name> dans les Exigences statiques du 7.5.

4.64 Section 11: erreur matérielle

Sans objet dans le texte français.

4.65 Section 11: erreur matérielle – Suppression des unités de temps dans les Figures 43, 44 et 46

La lettre "s" suivant certaines expressions de temps dans les Figures 43 et 44, ainsi que les lettres "ms" dans la Figure 46 doivent être supprimées. Elles représentent l'unité de temps, mais celle-ci ne

fait normalement pas partie des expressions dans la plupart des langages. De plus, ces symboles sont utilisés uniquement dans certaines expressions temporelles et pas les autres. Le texte accompagnant les figures doit aussi être changé, les indicateurs d'unité de temps devant être remplacés par les mots "unité de temps", qui ont été explicitement utilisés.

4.66 Section 11: erreur matérielle – Casse du mot clé msc dans les Figures 42 à 46

Le mot clé **MSC** qui apparaît dans les Figures 42 à 46 doit être en minuscules (**msc**).

4.67 Section 11: erreur matérielle – Adjonction de texte manquant dans la Figure 42

Les abréviations utilisées pour les paramètres associés aux messages sont incorrectes. Il convient d'utiliser les noms entiers.

4.68 Annexe A: erreur matérielle – Référence incorrecte à <keywords>

Dans la phrase introductive de l'Annexe A, supprimer "les mots clés <keyword> et" car les mots clés ne figurent pas dans l'index.

4.69 Point d'ordre général – Emploi de <end> ou du point virgule dans la grammaire textuelle

La production <end> est utilisée dans l'ensemble de la grammaire concrète alors qu'un point virgule serait plus précis. La production <end> permet un commentaire avant le point virgule et il a pour but d'être la version concrète du commentaire graphique. Toutefois, il y a beaucoup d'endroits où la grammaire concrète n'est pas obtenue à partir de la grammaire graphique mais qu'elle en est une partie. Autrement dit, la grammaire concrète est utilisée comme texte dans la partie graphique et un commentaire graphique ne pourrait donc jamais se produire. A de tels endroits, il serait préférable d'utiliser un point virgule plutôt que <end>. Cela n'est pas strictement nécessaire mais serait une clarification. Cette modification nécessite un examen soigneux de la grammaire pour situer tous les endroits où le changement devrait être fait; cela n'a pas été le cas jusqu'à présent.

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	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	RGT et maintenance des réseaux: systèmes de transmission, circuits téléphoniques, télégraphie, télécopie 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 de données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information et protocole Internet
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication