



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Z.100

Corrigendum 1
(10/2001)

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

Techniques de description formelle – Langage de
description et de spécification

SDL: Langage de description et de spécification
Corrigendum 1

Recommandation UIT-T Z.100 – 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	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.

SDL: Langage de description et de spécification

CORRIGENDUM 1

Résumé

Le présent corrigendum contient une liste de modifications de la Rec. UIT-T Z.100 (SDL) approuvée par la Commission d'études 10 en novembre 2000, des modifications supplémentaires approuvées par le groupe d'experts lors de réunions ultérieures ainsi que la liste finale approuvée par la Commission d'études 10 lors de la réunion de septembre 2001.

Source

Le Corrigendum 1 de la Recommandation Z.100 de l'UIT-T, élaboré par la Commission d'études 10 (2001-2004) de l'UIT-T, a été approuvé le 29 octobre 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

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

TABLE DES MATIÈRES

	Page
1 Objectifs et domaine d'application	1
2 Terminologie	1
3 Maintenance des Rec. UIT-T Z.100 à Z.109	1
4 Modifications de la Recommandation Z.100	2
4.1 Erreur d'écriture – Soulignement de <Natural literal name>	2
4.2 Extension – Ajouter un commentaire avant la parenthèse <left curly bracket>	2
4.3 Erreur d'écriture – Élément <end> – Annexe D	2
4.4 Erreur d'écriture – Mot clé d'instruction de boucle – Paragraphe 1.5: Différences....	2
4.5 Clarification – Paragraphe 5.3.2: Modèle	2
4.6 Erreur d'écriture et clarification – Paragraphe 5.4.2: BNF	2
4.7 Clarification – Exemple – Paragraphe 5.4.3: Métalanguage applicable à la grammaire graphique	3
4.8 Erreur d'écriture – Symboles de ligne – Paragraphe 5.4.3: Métalanguage applicable à la grammaire graphique	3
4.9 Modification – Syntaxe de commentaire – Paragraphe 6.1: Règles lexicales	3
4.10 Extension – Signe de domaine – Paragraphe 6.1: Règles lexicales	4
4.11 Erreur d'écriture – Mot clé d'instruction de boucle – Paragraphe 6.1: Règles lexicales	4
4.12 Erreur d'écriture – Référence – Paragraphe 6.3: Règles de visibilité, noms et identificateurs, Grammaire textuelle concrète	4
4.13 Erreur d'écritures – Subdivision des dessins – Paragraphe 6.6: Subdivision des dessins	4
4.14 Modification – Modification de la syntaxe de commentaire – Paragraphe 6.7: Commentaire	5
4.15 Erreur d'écriture – Grammaire de la règle <specification area> – Paragraphe 7.1: Cadre d'application	5
4.16 Erreur d'écriture – Utilisation de signaux vers un environnement pour des systèmes basés sur un type – Paragraphe 7.1: Cadre d'application	5
4.17 Clarification – Type avec instanciation potentielle – Paragraphe 7.1: Cadre d'application, Sémantique	6
4.18 Clarification – syntaxe graphique concrète pour <diagram in package> – Paragraphe 7.2: Paquetage	6
4.19 Erreur d'écriture – Grammaire pour <specification area> – Paragraphe 7.2: Paquetage, Grammaire textuelle concrète	6
4.20 Erreur d'écriture – Union de paquetages – Paragraphe 7.2: Paquetage, Modèle	6
4.21 Clarification – Paragraphe 8.1.1.1: Types d'agent, Grammaire abstraite	6
4.22 Clarification – Syntaxe simplifiée – Paragraphe 8.1.1.1: Types d'agent, Grammaire textuelle concrète	7

	Page
4.23 Clarification – Syntaxe simplifiée – Paragraphe 8.1.1.1: Types d'agent, Grammaire graphique concrète	7
4.24 Erreur d'écriture – Définition de l'ensemble de sortie complet –Paragraphe 8.1.1.1: Types d'agent, Sémantique	7
4.25 Clarification – Modèle de type d'agent – Paragraphe 8.1.1.1: Types d'agent, Modèle	7
4.26 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 8.1.1.2: Type de système, Grammaire textuelle concrète.....	8
4.27 Erreur d'écriture – Syntaxe simplifiée – Paragraphe 8.1.1.2: Type de système, Grammaire textuelle concrète.....	8
4.28 Erreur d'écriture – Accès omis/syntaxe simplifiée – Paragraphe 8.1.1.2: Type de système, Grammaire graphique concrète	8
4.29 Clarification – Paragraphe 8.1.1.2: Type de système, Grammaire graphique concrète.....	8
4.30 Clarification – Simplification de la syntaxe – Paragraphe 8.1.1.3: Type de bloc, Grammaire textuelle concrète.....	8
4.31 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 8.1.1.3: Type de bloc, Grammaire textuelle concrète.....	9
4.32 Clarification – Paragraphe 8.1.1.3: Type de bloc, Grammaire graphique concrète ...	9
4.33 Clarification – Paragraphe 8.1.1.3: Type de bloc, Grammaire graphique concrète ...	9
4.34 Correction d'erreur – Paragraphe 8.1.1.3: Type de bloc, Grammaire graphique concrète.....	9
4.35 Clarification – Simplification de la syntaxe – Paragraphe 8.1.1.4: Type de processus, Grammaire textuelle concrète	9
4.36 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 8.1.1.4: Type de processus, Grammaire textuelle concrète	9
4.37 Clarification – Paragraphe 8.1.1.4: Type de processus, Grammaire graphique concrète.....	9
4.38 Clarification – Paragraphe 8.1.1.4: Type de processus, Grammaire graphique concrète.....	10
4.39 Correction d'erreur – Paragraphe 8.1.1.4: Type de processus, Sémantique	10
4.40 Erreur d'écriture – Aligner avec l'Annexe F – Paragraphe 8.1.1.5: Type d'état composite, Grammaire abstraite	10
4.41 Correction d'erreur – Paragraphe 8.1.1.5: Type d'état composite, Grammaire abstraite.....	10
4.42 Clarification – Paragraphe 8.1.1.5: Type d'état composite, Grammaire textuelle concrète.....	10
4.43 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 8.1.1.5: Type d'état composite, Grammaire textuelle concrète.....	11
4.44 Clarification – Paragraphe 8.1.1.5: Type d'état composite, Grammaire textuelle concrète.....	11

	Page
4.45 Clarification – Paragraphe 8.1.1.5: Type d'état composite, Grammaire graphique concrète.....	11
4.46 Correction d'erreur – Paragraphe 8.1.1.5: Type d'état composite, Grammaire graphique concrète.....	11
4.47 Erreur d'écriture – Type souligné – Paragraphe 8.1.3.1: Définition de système fondée sur le type de système, Grammaire textuelle concrète	11
4.48 Erreur d'écriture – Transformation non nécessaire – Paragraphe 8.1.3.1: Définition de système fondée sur le type de système	12
4.49 Erreur d'écriture – Transformation non nécessaire – Paragraphe 8.1.3.2: Définition de bloc fondée sur le type de bloc	12
4.50 Erreur d'écriture – Transformation non nécessaire – Paragraphe 8.1.3.3: Définition de processus fondée sur le type de processus	13
4.51 Erreur d'écriture – Export absent – Paragraphe 8.1.4: Type abstrait, Grammaire concrète.....	13
4.52 Erreur d'écriture – Sorte dans une propriété de paramètre de signal – Paragraphe 8.1.5: Références de type, Grammaire textuelle concrète	13
4.53 Erreur d'écriture – Export absent – Paragraphe 8.1.5: Références de type, Grammaire graphique concrète	14
4.54 Erreur d'écriture – Contrainte de point d'extrémité trop restrictive – Paragraphe 8.1.6: Accès, Grammaire textuelle concrète.....	14
4.55 Erreur d'écriture – Identificateur erroné – Paragraphe 8.2.1: Paramètre de contexte de type d'agent, Grammaire textuelle concrète.....	14
4.56 Clarification – Signification du double point – Paragraphe 8.2.2: Paramètre de contexte d'agent, Grammaire textuelle concrète.....	14
4.57 Clarification – Lien vers la syntaxe abstraite – Paragraphe 8.3.1: Adjonction de propriétés, Grammaire textuelle concrète.....	14
4.58 Erreur d'écriture – Référence – Paragraphe 8.3.3: Transition/Sauvegarde virtuelle, Grammaire textuelle concrète.....	15
4.59 Erreur d'écriture – Référence à l'Annexe F – Paragraphe 8.3.3: Sémantique.....	15
4.60 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 8.4: Associations, Grammaire textuelle concrète.....	15
4.61 Clarification – Paragraphe 9: Agents, Grammaire abstraite.....	15
4.62 Clarification – Paragraphe 9: Agents, Grammaire textuelle concrète.....	15
4.63 Erreurs administratives – Paragraphe 9: Agents, Grammaire textuelle concrète	15
4.64 Clarification – Paragraphe 9: Agents, Grammaire textuelle concrète	16
4.65 Clarification – Simplification – Paragraphe 9: Agents, Grammaire textuelle concrète.....	16
4.66 Clarification – Mode de description – Paragraphe 9: Agents, Grammaire graphique concrète.....	16
4.67 Erreur d'écriture – Mode de description – Paragraphe 9: Agents, Grammaire graphique concrète.....	17

	Page
4.68 Erreur d'écriture – Restriction non nécessaire pour une connexion d'accès – Paragraphe 9: Agents, Sémantique.....	17
4.69 Clarification – Conditions d'arrêt pour des agents initiaux – Paragraphe 9: Agents, Sémantique	17
4.70 Erreur d'écriture – Ensemble de signaux d'entrée complets valides – Paragraphe 9: Agents, Sémantique.....	17
4.71 Erreur d'écriture – Pluriel incorrect – Paragraphe 9: Agents, Sémantique.....	18
4.72 Erreur d'écriture – Définition de l'ensemble de sortie complet – Paragraphe 9: Agents, Sémantique.....	18
4.73 Clarification – Valeur pid des agents contenus – Paragraphe 9: Agents, Modèle	18
4.74 Erreur d'écriture – Accès omis – Paragraphe 9.1: Système, Grammaire graphique concrète.....	18
4.75 Erreur d'écriture – Condition incorrecte – Paragraphe 9.1: Système, Grammaire graphique concrète.....	18
4.76 Clarification – Paragraphe 9.2: Bloc, Grammaire graphique concrète.....	18
4.77 Clarification – Paragraphe 9.3: Processus, Grammaire graphique concrète	19
4.78 Erreur d'écriture – Nœud de départ/Identificateur de procédure de base – Paragraphe 9.4: Procédure, Grammaire abstraite	19
4.79 Erreur d'écriture – Export absent – Paragraphe 9.4: Procédure, Grammaire textuelle concrète.....	19
4.80 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 9.4: Procédure, Grammaire textuelle concrète.....	19
4.81 Clarification – Paragraphe 9.4: Procédure, Grammaire graphique concrète.....	19
4.82 Erreur d'écriture – Etat composite absent – Paragraphe 9.4: Procédure, Grammaire graphique concrète.....	20
4.83 Clarification – Paragraphe 9.4: Procédure, Grammaire graphique concrète.....	20
4.84 Clarification – Paragraphe 9.4: Procédure, Grammaire graphique concrète.....	20
4.85 Erreur d'écriture – Paragraphe 9.4: Procédure, Sémantique.....	20
4.86 Erreur d'écriture – Référence à l'Annexe F – Paragraphe 9.4: Procédure, Sémantique	20
4.87 Erreur d'écriture – Référence à l'Annexe F – Paragraphe 10.1: Canal, Grammaire textuelle concrète.....	21
4.88 Clarification – Mode de description – Paragraphe 10.1: Canal, Grammaire graphique concrète.....	21
4.89 Clarification – Mode de description – Paragraphe 10.1: Canal, Grammaire graphique concrète.....	21
4.90 Erreur d'écriture – Référence à l'Annexe F – Paragraphe 10.1: Canal, Modèle.....	22
4.91 Erreur d'écriture – Paragraphe 10.2: Connexion, Grammaire graphique concrète.....	23
4.92 Erreur d'écriture – Paragraphe 10.2: Connexion, Modèle	24

	Page
4.93 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 10.3: Signal, Grammaire textuelle concrète.....	24
4.94 Clarification – Paragraphe 10.3: Signal, Grammaire graphique concrète.....	24
4.95 Erreur d'écriture – Interface absente – Paragraphe 10.4: Définition de liste de signaux, Grammaire textuelle concrète	24
4.96 Erreur d'écriture – Interface absente – Paragraphe 10.4: Définition de liste de signaux, Grammaire textuelle concrète	24
4.97 Erreur d'écriture – Exception absente – Paragraphe 10.5: Procédures distantes, Grammaire textuelle concrète.....	25
4.98 Erreur d'écriture – Définition de l'ensemble de sortie complet – Paragraphe 10.5: Procédures distantes, Grammaire textuelle concrète.....	25
4.99 Erreur d'écriture – Définition de l'ensemble de sortie complet – Paragraphe 10.5: Procédures distantes, Grammaire textuelle concrète.....	25
4.100 Erreur d'écriture – Retour supplémentaire, tabulation supplémentaire – Paragraphe 10.5: Procédures distantes, Modèle	25
4.101 Erreur d'écriture – Nodelay- Paragraphe 10.5: Procédures distantes, Modèle	25
4.102 Erreur d'écriture – Exception absente – Paragraphe 10.6: Variables distantes, Grammaire textuelle concrète.....	26
4.103 Erreur d'écriture – Référence à l'Annexe F et nodelay – Paragraphe 10.6: Variables distantes.....	26
4.104 Erreur d'écriture – Erreur de présentation – Paragraphe 11.2: Etat, Grammaire abstraite.....	26
4.105 Clarification – Présentation d'état de base et d'état composite – Paragraphe 11.2: Etat, Grammaire abstraite	26
4.106 Clarification – Présentation d'état de base et d'état composite – Paragraphe 11.2: Etat, Grammaire textuelle concrète	27
11.2.1 Etat de base.....	27
4.107 Clarification – Présentation d'état de base et d'état composite – Paragraphe 11.2: Etat, Grammaire graphique concrète.....	28
4.108 Clarification – Présentation d'état de base et d'état composite – Paragraphe 11.2: Etat, Sémantique.....	28
4.109 Clarification – Sélection de transition – Paragraphe 11.2.1: Etat de base, Sémantique	28
4.110 Clarification – Présentation d'état de base et d'état composite – Paragraphe 11.2.1: Etat de base, Modèle.....	29
4.111 Clarification – Remplacer <action statement> par <action> – Paragraphe 11.3: Entrée, Modèle	29
4.112 Erreur d'écriture – On-exception omis – Paragraphe 11.5: Signal continu, Grammaire abstraite	29
4.113 Clarification – Sélection de transition – Paragraphe 11.5: Signal continu, Sémantique	29

	Page
4.114 Clarification – Sélection de transition – Paragraphe 11.6: Condition de validation, Sémantique	30
4.115 Clarification – Paragraphe 11.7: Sauvegarde, Grammaire abstraite	30
4.116 Clarification – Zone de corps à la place de zone de graphe – Paragraphe 11.10: Etiquette, Grammaire textuelle concrète	30
4.117 Clarification – Remplacer <action statement> par <action> – Paragraphe 11.10: Etiquette, Grammaire textuelle concrète	30
4.118 Clarification – Remplacer "graph area" par "body area" – Paragraphe 11.10: Etiquette, Grammaire graphique concrète	30
4.119 Clarification – Remplacer <action statement> par <action> – Paragraphe 11.10: Etiquette, Modèle	30
4.120 Clarification – Remplacer <terminator statement> par <terminator> – Paragraphe 11.10: Etiquette, Modèle	31
4.121 Erreur d'écriture – Mot "by" absent – Paragraphe 11.11: Machine à états et état composite.....	31
4.122 Clarification – Paragraphe 11.11: Machine à états et état composite, Sémantique....	31
4.123 Erreur d'écriture – Modèle absent – Paragraphe 11.11: Machine à états et état composite, Modèle	31
4.124 Erreur d'écriture – Nœud de départ optionnel – Paragraphe 11.11.1: Graphe d'état composite, Grammaire abstraite	31
4.125 Clarification – Paragraphe 11.11.1: Graphe d'état composite, Grammaire textuelle concrète.....	32
4.126 Erreur d'écriture – Spécialisation omise – Paragraphe 11.11.1: Graphe d'état composite, Grammaire textuelle concrète	32
4.127 Erreur d'écriture – Eléments omis – Paragraphe 11.11.1: Graphe d'état composite, Grammaire textuelle concrète.....	32
4.128 Erreur d'écriture – Règle mal placée – Paragraphe 11.11.1: Graphe d'état composite, Grammaire textuelle concrète	32
4.129 Clarification – Phrase mal placée – Paragraphe 11.11.1: Graphe d'état composite, Grammaire textuelle concrète.....	32
4.130 Clarification – Utilisation d'accès – Paragraphe 11.11.1: Graphe d'état composite, Grammaire textuelle concrète.....	33
4.131 Clarification – Paragraphe 11.11.1: Graphe d'état composite, Grammaire graphique concrète.....	33
4.132 Erreur d'écriture – Eléments omis – Paragraphe 11.11.1: Graphe d'état composite, Grammaire graphique concrète	33
4.133 Clarification – Paragraphe 11.11.1: Graphe d'état composite, Grammaire graphique concrète.....	33
4.134 Clarification – Phrase mal placée – Paragraphe 11.11.1: Graphe d'état composite, Grammaire graphique concrète	34
4.135 Clarification – Paragraphe 11.11.1: Graphe d'état composite, Sémantique	34
4.136 Clarification – Paragraphe 11.11.2: Agrégat d'état	34

	Page
4.137 Clarification – Conditions – Paragraphe 11.11.2: Agrégat d'état, Grammaire abstraite.....	34
4.138 Clarification – Paragraphe 11.11.2: Agrégat d'état, Grammaire abstraite.....	35
4.139 Clarification – Paragraphe 11.11.2: Agrégat d'état, Grammaire textuelle concrète ...	35
4.140 Erreur d'écriture – Spécialisation omise – Paragraphe 11.11.2: Agrégat d'état, Grammaire textuelle concrète.....	35
4.141 Clarification – Paragraphe 11.11.2: Agrégat d'état, Grammaire textuelle concrète...	35
4.142 Erreur d'écriture – Autoriser un corps vide – Paragraphe 11.11.2: Agrégat d'état, Grammaire textuelle concrète.....	35
4.143 Clarification – Référence d'état composite – Paragraphe 11.11.2: Agrégat d'état, Grammaire textuelle concrète.....	35
4.144 Erreur d'écriture – Référence d'état composite – Paragraphe 11.11.2: Agrégat d'état, Grammaire textuelle concrète	35
4.145 Clarification – Paragraphe 11.11.2: Agrégat d'état, Grammaire graphique concrète.	36
4.146 Erreur d'écriture – Autoriser un corps vide – Paragraphe 11.11.2: Agrégat d'état, Grammaire graphique concrète	36
4.147 Clarification – Paragraphe 11.11.2: Agrégat d'état, Grammaire graphique concrète.	36
4.148 Erreur d'écriture – Identificateur – Paragraphe 11.11.2: Agrégat d'état, Grammaire graphique concrète.....	36
4.149 Erreur d'écriture – Syntaxe de points de connexion – Paragraphe 11.11.2: Agrégat d'état, Grammaire graphique concrète.....	36
4.150 Erreur d'écriture – Accès omis – Paragraphe 11.11.2: Agrégat d'état, Grammaire graphique concrète.....	37
4.151 Clarification – Conditions – Paragraphe 11.11.2: Agrégat d'état, Grammaire graphique concrète.....	37
4.152 Clarifications – Paragraphe 11.11.2: Agrégat d'état, Sémantique	37
4.153 Clarification – Phrase mal placée – Paragraphe 11.11.3: Point de connexion d'état, Sémantique	37
4.154 Extension – Liste de connexions plus flexible – Paragraphe 11.11.4: Connexion.....	38
4.155 Erreur d'écriture – Traitement d'exception – Paragraphe 11.12.1: Corps de transition, Grammaire textuelle concrète	38
4.156 Erreur d'écriture – Corps d'opérateurs – Paragraphe 11.12.1: Corps de transition, Grammaire graphique concrète	39
4.157 Erreur d'écriture – Référence à l'Annexe F – Paragraphe 11.12.1: Corps de transition, Sémantique	39
4.158 Clarification – Remplacer <action statement> par <action> – Paragraphe 11.12.2.2: Branchement.....	39
4.159 Clarification – <agent body > – Paragraphe 11.12.2.2: Branchement, Grammaire textuelle concrète.....	39
4.160 Clarification – <agent body area> – Paragraphe 11.12.2.2: Branchement, Grammaire graphique concrète	40

	Page
4.161 Extension – Autoriser un arrêt dans les procédures, etc. – Paragraphe 11.12.2.3: Arrêt.....	40
4.162 Erreur d'écriture – Sorte de retour – Paragraphe 11.12.2.4: Retour, Grammaire abstraite.....	40
4.163 Erreur d'écritures – Cohérence entre les représentations PR et GR – Paragraphe 11.12.2.4: Retour, Grammaires concrètes	40
4.164 Erreur d'écriture – Sorte de retour – Paragraphe 11.12.2.4: Retour, Grammaire graphique concrète.....	40
4.165 Erreur d'écriture – Sorte de retour – Paragraphe 11.12.2.4: Retour, Sémantique	41
4.166 Erreur d'écriture – Traitement d'exception – Paragraphe 11.13.1: Tâche, Grammaire textuelle concrète.....	41
4.167 Clarification et modification – Paragraphe 11.13.1: Tâche.....	41
4.168 Erreur d'écriture – Traitement d'exception – Paragraphe 11.13.2: Création, Grammaire textuelle concrète.....	42
4.169 Clarification – Détermination statique d'ensembles d'instances – Paragraphe 11.13.2: Création, Modèle	42
4.170 Erreur d'écriture – Incohérence opération/procédure – Paragraphe 11.13.3: Appel de procédure, Grammaire abstraite.....	42
4.171 Erreur d'écriture – Traitement d'exception – Paragraphe 11.13.3: Appel de procédure, Grammaire textuelle concrète.....	43
4.172 Erreur d'écriture – Incohérence opération/procédure – Paragraphe 11.13.3: Appel de procédure, Sémantique	43
4.173 Erreur d'écriture – Traitement d'exception – Paragraphe 11.13.4: Sortie, Grammaire textuelle concrète.....	43
4.174 Extension – Types de donnée en ligne – Paragraphe 11.13.4: Sortie, Grammaire textuelle concrète.....	43
4.175 Correction d'erreur – Evaluation de réponses de décision – Paragraphe 11.13.5: Décision	44
4.176 Clarification – Remplacer <terminator/action statement> par <terminator/action> – Paragraphe 11.13.5: Décision, Grammaire textuelle concrète.....	44
4.177 Correction d'erreur – Evaluation de réponses de décision – Paragraphe 11.13.5: Décision.....	44
4.178 Erreur d'écriture – Syntaxe erronée – Paragraphe 11.13.5: Décision, Modèle.....	44
4.179 Extension – Autoriser dans les procédures, etc. – Paragraphe 11.14: Liste d'instructions, Grammaire textuelle concrète	44
4.180 Erreur d'écriture – Aligner avec l'Annexe F – Paragraphe 11.14.1: Instruction composée, Grammaire abstraite	45
4.181 Erreur d'écriture – Modèle incorrect – Paragraphe 11.14.1: Instruction composée, Modèle.....	45

	Page
4.182 Correction d'erreur – Autoriser des procédures distantes – Paragraphe 11.14.2: Actions et terminateurs de transition en tant qu'instructions, Grammaire textuelle concrète.....	45
4.183 Clarification – Remplacer <action statement> par <action> – Paragraphe 11.14.2: Actions et terminateurs de transition en tant qu'instructions.....	45
4.184 Erreur d'écriture – Traitement d'exception – Paragraphe 11.14.2: Actions et terminateurs de transition en tant qu'instructions transition, Grammaire textuelle concrète.....	45
4.185 Extension – Autoriser des arrêts dans les procédures, etc. – Paragraphe 11.14.2: Actions et terminateurs de transition en tant qu'instructions.....	46
4.186 Clarification – Paragraphe 11.14.4: Instruction Si, Modèle.....	46
4.187 Erreur d'écriture – Paragraphe 11.14.5: Instruction de décision	46
4.188 Erreur d'écriture – Mot clé – Paragraphe 11.14.6: Instruction de boucle, Grammaire textuelle concrète.....	46
4.189 Erreur d'écriture – Paragraphe 11.14.9: Instruction d'exception, Grammaire textuelle concrète.....	47
4.190 Erreur d'écriture – Traitement d'exception – Paragraphe 11.15: Temporisateur, Grammaire textuelle concrète.....	47
4.191 Erreur d'écriture – Traitement d'exception – Paragraphe 11.16.2: On-exception, Grammaire textuelle concrète.....	47
4.192 Clarification – Remplacer <action statement> par <action> – Paragraphe 11.16.3: Traitement, Modèle	47
4.193 Erreur d'écriture – Parties absentes, aligner avec l'Annexe F – Paragraphe 12.1: Définition de données, Grammaire abstraite	47
4.194 Extension – Types de donnée en ligne – Paragraphe 12.1: Définitions de données, Grammaire textuelle concrète.....	48
4.195 Correction d'erreur – Valeur ou objet pour une sorte ancrée – Paragraphe 12.1: Définitions de données, Grammaire textuelle concrète.....	48
4.196 Extension – Types de donnée en ligne – Paragraphe 12.1: Définitions de données, Modèle.....	48
4.197 Clarification – Paragraphe 12.1.1: Définition de type de données.....	49
4.198 Erreur de syntaxe – Élément <end> – Paragraphe 12.1.1: Définitions des types de données, Grammaire textuelle concrète.....	49
4.199 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 12.1.1: Définitions des types de données, Grammaire textuelle concrète.....	49
4.200 Clarification – Paragraphe 12.1.1: Définitions des types de données, Grammaire textuelle concrète.....	50
4.201 Clarification – Paragraphe 12.1.1: Définitions des types de données, Grammaire graphique concrète.....	50
4.202 Erreur d'écriture – Barre verticale superflue – Paragraphe 12.1.2: Définition d'interface, Grammaire textuelle concrète.....	50

	Page
4.203 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 12.1.2: Définition d'interface, Grammaire textuelle concrète.....	50
4.204 Clarification – Paragraphe 12.1.2: Définition d'interface, Grammaire graphique concrète.....	50
4.205 Clarification – Interfaces – Paragraphe 12.1.2: Définition d'interface, Sémantique..	51
4.206 Clarification – Interface implicite – Paragraphe 12.1.2: Définition d'interface, Modèle.....	51
4.207 Erreur d'écriture – Aligner avec l'Annexe F – Paragraphe 12.1.4: Opérations, Grammaire abstraite	52
4.208 Erreur d'écriture – Ambiguïté de syntaxe – Paragraphe 12.1.4: Opérations, Grammaire textuelle concrète.....	52
4.209 Erreur d'écriture – Emplacement du texte – Paragraphe 12.1.5: Quelconque (Any) .	52
4.210 Clarification – Interfaces implicites – Paragraphe 12.1.6: Pid et sortes de pid, Sémantique	52
4.211 Erreur d'écriture – Paragraphe 12.1.7.3: Type de données de choix, Modèle.....	52
4.212 Erreur d'écriture – Sortes optionnelles – Paragraphe 12.1.8: Comportement des opérations, Grammaire graphique concrète.....	52
4.213 Erreur d'écriture – Paragraphe 12.1.8: Comportement des opérations, Sémantique ..	53
4.214 Extension – Signe de domaine – Paragraphe 12.1.9.1: Classe de nom, Grammaire textuelle concrète.....	53
4.215 Erreur de syntaxe – Élément <end> – Paragraphe 12.1.9.4: Syntypes, Grammaire textuelle concrète.....	53
4.216 Erreur d'écriture – Variante omise – Paragraphe 12.1.9.4: Syntypes, Grammaire textuelle concrète.....	53
4.217 Modification – Difficulté grammaticale – Paragraphe 12.1.9.4: Syntypes, Grammaire textuelle concrète.....	53
4.218 Erreur d'écriture – Ambiguïté – Paragraphe 12.1.9.4: Syntypes, Grammaire textuelle concrète.....	54
4.219 Extension – Signe de domaine – Paragraphe 12.1.9.5: Contrainte, Grammaire textuelle concrète.....	54
4.220 Erreur d'écriture – Paragraphe 12.1.9.6: Définition de synonyme, Grammaire textuelle concrète.....	54
4.221 Erreur d'écriture – Paragraphe 12.2.1: Expressions, Grammaire abstraite.....	54
4.222 Extension – Types de donnée en ligne – Paragraphe 12.2.1: Expressions, Grammaire textuelle concrète.....	54
4.223 Erreur d'écriture – Paragraphe 12.2.1: Expressions, Grammaire textuelle concrète ..	55
4.224 Erreur d'écriture – Phrase fausse – Paragraphe 12.2.7: Application d'opération, Grammaire textuelle concrète.....	55
4.225 Clarification – Traitement de paramètre – Paragraphe 12.2.7: Application d'opération, Sémantique	55

	Page
4.226 Correction d'erreur – Restriction de paramètres – Paragraphe 12.2.7: Application d'opération, Modèle	55
4.227 Correction d'erreur – Règles trop restrictives pour l'initialisation des données – Paragraphe 12.3.1: Définition de variable	56
4.228 Erreur de syntaxe – Élément <end> – Paragraphe 12.3.3.2: Initialisation par défaut, Grammaire textuelle concrète	56
4.229 Erreur d'écriture – Paragraphe 12.3.4.6: Expression d'état	56
4.230 Erreur d'écriture – Éléments omis et dupliqués – Paragraphe 13.1: Définition optionnelle, Grammaire textuelle concrète	57
4.231 Clarification – Paragraphe 13.1: Définition optionnelle, Grammaire textuelle concrète	57
4.232 Clarification – Remplacer <action statement> par <action> – Paragraphe 13.2: Chaîne de transition optionnelle, Modèle	57
4.233 Clarification – Remplacer <terminator statement> par <terminator> – Paragraphe 13.2: Chaîne de transition optionnelle, Modèle	57
4.234 Erreur d'écriture – Mot clé de quantification – Paragraphe D.2.1: Axiomes, Grammaire textuelle concrète	57
4.235 Clarification – Supprimer la quantification optionnelle – Paragraphe D.2.1: Axiomes, Grammaire textuelle concrète	57
4.236 Clarification – Texte informel non utilisé – Paragraphe D.2.1: Axiomes, Sémantique	58
4.237 Clarification – Supprimer la quantification optionnelle – Paragraphe D.2.1: Axiomes, Sémantique	58
4.238 Clarification – Case des noms d'opérateurs – Paragraphe D.3: Paquetage prédéfini (Predefined)	58
4.239 Erreur d'écriture – Mot clé inherit absent – Paragraphe D.3.13.2: Utilisation	58
4.240 Erreur d'écriture – Mots clés en majuscules et minuscules – Appendice III	58
4.241 Clarifications – Appendice III	58
4.242 Clarifications – Appendice III	58

Recommandation UIT-T Z.100

SDL: Langage de description et de spécification

CORRIGENDUM 1

1 Objectifs et domaine d'application

L'objectif du présent corrigendum est de prendre acte des modifications qui ont fait l'objet d'un accord pour la Rec. UIT-T Z.100 (1999).

Les modifications appartiennent aux deux catégories suivantes:

- a) Correction d'*erreurs* et *clarifications* (se référer aux définitions données dans l'Appendice II/Z.100).
- b) *Extensions* et *modifications* (se référer aux définitions données dans l'Appendice II/Z.100).

Les règles de maintenance données dans l'Appendice II/Z.100 stipulent que les *erreurs* et les *clarifications* publiées dans la liste principale de modifications "prennent effet immédiatement". De telles modifications doivent faire l'objet d'une publication dans un corrigendum aussi rapidement que possible.

Les *modifications* et les *extensions* impliquent certaines modifications du langage SDL. La règle suivante s'applique dans ce cas: "Sauf circonstances spéciales nécessitant l'implémentation aussi rapide que possible de telles modifications, ces dernières ne seront pas recommandées tant que la Rec. Z.100 n'aura pas été révisée". Les *modifications* et les *extensions* contenues dans le présent corrigendum ne sont pas considérées comme significatives, mais on estime qu'elles fournissent des améliorations utiles pour la notation SDL et on recommande, de ce fait, leur utilisation immédiate.

2 Terminologie

Une *erreur* est une incohérence dans une ou plusieurs des Rec. UIT-T Z.100 à Z.109.

Une *correction textuelle* est une modification du texte ou des diagrammes de Recommandations qui corrigent des erreurs de bureautique ou de typographie.

Un *point ouvert* correspond à un problème ouvert mais non résolu.

Une *déficience* correspond à un problème identifié pour lequel la syntaxe SDL n'est pas définie clairement dans les Recommandations.

Une *clarification* correspond à une modification d'un texte (ou de diagrammes) dans une Recommandation qui ne modifie pas (de manière intentionnelle) la signification du diagramme SDL, mais qui est prévue pour rendre les Recommandations moins ambiguës ou plus compréhensibles.

Une *modification* change la sémantique du langage SDL.

Une *extension* correspond à une nouvelle fonctionnalité qui ne modifie pas la sémantique du langage SDL défini dans les Recommandations approuvées pour ce langage SDL.

3 Maintenance des Rec. UIT-T Z.100 à Z.109

L'Appendice II/Z.100 (1999) indique la procédure à appliquer pour la maintenance des Rec. Z.100. Cette procédure nécessite une large publication des corrections d'erreur, des propositions de modification et d'extension, ainsi que la gestion d'une liste maîtresse des modifications. Bien que le

présent corrigendum contienne une liste de modifications, la liste maîtresse des modifications contient également d'autres informations qui ne concernent pas le présent corrigendum.

4 Modifications de la Recommandation Z.100

4.1 Erreur d'écriture – Soulignement de <Natural literal name>

Remplacer toutes les occurrences de <Natural literal name> par <Natural literal name>.

Ces occurrences apparaissent dans les règles de syntaxe <number of pages>, <priority name> et <partial regular expression> ainsi que dans la description de la règle <partial regular expression>.

4.2 Extension – Ajouter un commentaire avant la parenthèse <left curly bracket>

Ajouter l'élément [<comment body>] avant chaque occurrence de la règle <left curly bracket> dans les règles de syntaxe, à l'exception des règles lexicales: c'est-à-dire dans les § 9.4, 11.13.1, 11.14.1, 11.14.5, 12.1, 12.1.1, 12.1.2, 12.1.8 et 12.1.9.4 (plusieurs fois dans certains paragraphes).

4.3 Erreur d'écriture – Élément <end> – Annexe D

Dans l'Annexe D, les en-têtes <data type heading> doivent tous se terminer par un élément <end>.

4.4 Erreur d'écriture – Mot clé d'instruction de boucle – Paragraphe 1.5: Différences ...

Le mot clé "for" doit être remplacé par "loop".

Ajouter "loop" à la liste des mots clés du langage SDL-2000 qui ne sont pas des mots clés du langage SDL-92.

Ajouter "for" à la liste des mots clés du langage SDL-92 qui ne sont pas des mots clés du langage SDL-2000.

4.5 Clarification – Paragraphe 5.3.2: Modèle

Ajouter le texte suivant à la fin du paragraphe *Modèle*:

"Les transformations peuvent être interdépendantes, ce qui fait que l'ordre d'application des diverses transformations peut influencer sur la validité et la signification d'une spécification SDL. Les Annexes F1 à F3 fournissent les détails précis concernant l'ordre des transformations."

4.6 Erreur d'écriture et clarification – Paragraphe 5.4.2: BNF

Modifier les quatre lignes suivantes de l'exemple de syntaxe:

```
<diagram in package> ::=  
    | <package diagram>  
    | <package reference area>  
    | <type in agent area>
```

en supprimant les barres verticales erronées et renommer la zone <type in agent area> de manière à obtenir la clarification suivante:

```
    <package diagram>  
    | <package reference area>  
    | <entity in agent diagram>
```

Remplacer "<type in agent area>" par "<entity in agent diagram>" dans l'alinéa qui fait suite à l'exemple, de manière à lire comme suit:

"indique qu'un diagramme <diagram in package> est un diagramme <package diagram>, une zone <package reference area>, un diagramme <entity in agent diagram> ou une zone <data type reference area>, ..."

4.7 Clarification – Exemple – Paragraphe 5.4.3: Métalangage applicable à la grammaire graphique

Pour tenir compte d'une clarification de la syntaxe réelle qui remplace la règle <operation graph area> par la règle <operation body area>, modifier comme suit l'exemple et le texte qui lui fait suite:

{ <operation text area>* <operation body area> } **set**

est un ensemble de zéro, d'une ou de plusieurs zones <operation text area> et d'une zone <operation body area>. Le métasymbole **set** ..."

4.8 Erreur d'écriture – Symboles de ligne – Paragraphe 5.4.3: Métalangage applicable à la grammaire graphique

Le texte "et implique un flow line symbol (voir 6.5)" figure dans un alinéa incorrect. Remplacer les deux derniers alinéas:

"Le métasymbole **is followed by** signifie que son argument de droite suit (tant sur le plan logique que dans le dessin) son argument de gauche.

Le métasymbole **is connected to** signifie que son argument de droite est relié (tant sur le plan logique que dans le dessin) à son argument de gauche et implique un flow line symbol (voir 6.5)."

par les alinéas suivants:

"Le métasymbole **is followed by** signifie que son argument de droite suit (tant sur le plan logique que dans le dessin) son argument de gauche et implique un flow line symbol (voir 6.5).

Le métasymbole **is connected to** signifie que son argument de droite est relié (tant sur le plan logique que dans le dessin) à son argument de gauche."

4.9 Modification – Syntaxe de commentaire – Paragraphe 6.1: Règles lexicales

La syntaxe de commentaire <comment > est remplacée par **comment** /# ... #/ de manière à autoriser également l'utilisation de /# ... #/ comme commentaire avant un délimiteur "{". Les modifications suivantes sont faites à cet effet:

Le corps <comment body> est ajouté comme variante dans l'unité <lexical unit>

<lexical unit> ::=

	<name>
	<quoted operation name>
	<character string>
	<hex string>
	<bit string>
	<note>
	<comment body>
	<composite special>
	<special>
	<keyword>

Il convient d'introduire l'élément <number sign> comme variante dans les règles suivantes du fait qu'il est supprimé dans la règle <other special>:

<note text> ::=

```
{
|   <general text character>
|   <other special>
|   <number sign>
|   <asterisk>+ <not asterisk or solidus>
|   <solidus>
|   <apostrophe> }*
```

<not asterisk or solidus> ::=

<general text character> | <other special> | <apostrophe> | <number sign>

<special> ::=

<solidus> | <asterisk> | <number sign> | <other special>

Définir <comment body> au moyen des règles suivantes:

<comment body> ::=

<solidus> <number sign> <note text> <number sign>+ <solidus>

<comment text> ::=

```
{
|   <general text character>
|   <other special>
|   <asterisk>
|   <number sign>+ <not number or solidus>
|   <solidus>
|   <apostrophe> }*
```

<not number or solidus> ::=

<general text character> | <other special> | <apostrophe> | <asterisk>

4.10 Extension – Signe de domaine – Paragraphe 6.1: Règles lexicales

Modifier la règle <composite special> de manière à inclure la variante <range sign> – c'est-à-dire ajouter une nouvelle ligne après l'élément <result sign>

| <range sign>

Ajouter la règle suivante après la règle <result sign>:

<range sign> ::=

<full stop> <full stop>

4.11 Erreur d'écriture – Mot clé d'instruction de boucle – Paragraphe 6.1: Règles lexicales

Le mot clé **for** doit être remplacé par **loop**. Supprimer "**for**" dans la règle <keyword> et ajouter "**loop**". Remettre en forme la liste des mots.

4.12 Erreur d'écriture – Référence – Paragraphe 6.3: Règles de visibilité, noms et identificateurs, Grammaire textuelle concrète

Dans l'alinéa:

"Une définition de référence est une entité après l'étape de transformation de la définition <referenced definition> (voir Annexe F)."

Remplacer la référence à l'"Annexe F" par une référence au paragraphe "7.3".

4.13 Erreur d'écritures – Subdivision des dessins – Paragraphe 6.6: Subdivision des dessins

Le mot "implicit" doit être souligné dans la règle:

<heading area> ::=

<implicit text symbol> **contains** <heading>

Le mot "drawing" doit être souligné dans la règle:

<kernel heading> ::=
 [<virtuality>] [**exported**]
 <drawing kind> <drawing qualifier> <drawing name>

Insérer une barre verticale manquante en début de ligne:

 | **state** [**type**] | **procedure** | **operator** | **method**

Le mot "implicit" doit être souligné dans l'expression:

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

NOTE – Seul, le mot "Natural" doit être souligné dans la règle <number of pages> (se référer à la modification globale).

Le mot "implicit" doit être souligné dans "<implicit text symbol>" dans l'alinéa débutant par:

"Afin d'avoir une séparation nette entre la zone <heading area> et la zone <page number area>, le symbole <implicit text symbol> n'est pas matérialisé, il est sous-entendu."

4.14 Modification – Modification de la syntaxe de commentaire – Paragraphe 6.7: Commentaire

La syntaxe est modifiée comme suit:

<comment> ::=
 comment <comment body>

4.15 Erreur d'écriture – Grammaire de la règle <specification area> – Paragraphe 7.1: Cadre d'application

Il semble qu'il y ait eu un problème d'édition pour ce paragraphe dont la grammaire n'est pas tout à fait correcte. La correction suivante s'applique de ce fait.

Le début de la règle <specification area>:

<specification area> ::=
 <frame symbol> **contains**

est modifié comme suit en ajoutant une accolade ouvrante:

<specification area> ::=
 <frame symbol> **contains** {

4.16 Erreur d'écriture – Utilisation de signaux vers un environnement pour des systèmes basés sur un type – Paragraphe 7.1: Cadre d'application

La syntaxe est modifiée de manière à autoriser des paquetages pour des systèmes basés sur un type, ce qui permet de définir, pour un tel système, des signaux vers l'environnement. Les règles syntactiques <textual system specification> et <graphical system specification> sont modifiées comme suit:

Grammaire textuelle concrète

<textual system specification> ::=
 <agent definition>
 | {<package use clause>}* <textual typebased agent definition>

Grammaire graphique concrète

<graphical system specification> ::=
 <agent diagram>
 | <graphical typebased agent definition> [**is associated with** <package use area>]

et le texte suivant est ajouté à la fin du paragraphe *Modèle*:

"Une clause <package use clause> figurant avant une définition <textual typebased agent definition> dans une règle <textual system specification> est une syntaxe dérivée pour une clause <package use clause> figurant avant l'en-tête <system heading> de la définition <system definition> dérivée de la définition <textual typebased agent definition>.

Une zone <package use area> associée à une définition <graphical typebased agent definition> dans une règle <graphical system specification> est une syntaxe dérivée pour une zone <package use area> associée au diagramme <system diagram> dérivé de la définition <graphical typebased agent definition>."

4.17 Clarification – Type avec instanciation potentielle – Paragraphe 7.1: Cadre d'application, Sémantique

Ajouter l'alinéa suivant à la fin du paragraphe *Sémantique*:

"Dans le cas d'une spécification *SDL-specification* avec une définition *Agent-definition*, un type est *instancié de manière potentielle* s'il est instancié dans la définition *Agent-definition* ou s'il est instancié dans un type potentiellement instanciable."

4.18 Clarification – Syntaxe graphique concrète pour <diagram in package> – Paragraphe 7.2: Paquetage

La variante "<type in agent area>" est renommée "<entity in agent diagram>".

4.19 Erreur d'écriture – Grammaire pour <specification area> – Paragraphe 7.2: Paquetage, Grammaire textuelle concrète

Modifier le texte qui suit la syntaxe de la règle <dependency symbol>:

"... pour le diagramme <package diagram> ou du progiciel <package> ou de la spécification <system specification> pour une zone <package reference area> contenue dans une zone <specification area>, et ces zones doivent être cohérentes avec cette clause <package use clause>"

par:

"... pour le diagramme <package diagram> (ou du progiciel <package> ou de la spécification <system specification> pour une zone <package reference area> contenue dans une zone <specification area>) et ces zones doivent être cohérentes avec cette clause <package use clause>."

4.20 Erreur d'écriture – Union de paquetages – Paragraphe 7.2: Paquetage, Modèle

Supprimer l'alinéa:

"Si un paquetage est mentionné dans plusieurs clauses <package use clause> d'une définition <package definition>, cela correspond à une clause <package use clause> qui sélectionne l'union des définitions choisies dans les clauses <package use clause>."

4.21 Clarification – Paragraphe 8.1.1.1: Types d'agent, Grammaire abstraite

Les définitions concernant *Agent-kind*, *Agent-formal-parameter*, *State-machine-definition* ont été retirées de ce paragraphe et déplacées vers le paragraphe 9.

Ajouter *Agent-kind* avant *Agent-type-identifier* et *Agent-formal-parameter*, *State-machine-definition* après *Agent-type-identifier* de manière à obtenir les définitions suivantes:

<i>Agent-kind</i>	=	SYSTEM BLOCK PROCESS
<i>Agent-type-identifier</i>	=	<i>Identifier</i>
<i>Agent-formal-parameter</i>	=	<i>Parameter</i>

State-machine-definition :: *State-name*
Composite-state-type-identifier

4.22 Clarification – Syntaxe simplifiée – Paragraphe 8.1.1.1: Types d'agent, Grammaire textuelle concrète

Dans la règle <agent type structure> remplacer "<state partitioning>" par "<state partition>".

Les définitions <system type definition>, <block type definition> et <process type definition> sont toutefois affinées sous la forme d'une structure <agent structure> plutôt que d'une structure <agent type structure>, les règles <agent type structure> et <agent type body> ne sont pas nécessaires de ce fait et la syntaxe est simplifiée.

4.23 Clarification – Syntaxe simplifiée – Paragraphe 8.1.1.1: Types d'agent, Grammaire graphique concrète

Remplacer le nom de la règle "<agent type diagram content>" par "<agent type structure area>".

Comme pour la grammaire textuelle, les éléments <agent type diagram content>, <type in agent area>, <agent type body area> et <type state machine graph area> sont supprimés parce que les diagrammes <system type diagram>, <block type diagram> et <process type diagram> sont définis en termes de zone <agent structure area>.

4.24 Erreur d'écriture – Définition de l'ensemble de sortie complet – Paragraphe 8.1.1.1: Types d'agent, Sémantique

Insérer le texte suivant à la fin du paragraphe *Sémantique*:

"L'ensemble de sortie complet d'un type d'agent est l'union de tous les signaux, procédures distantes et variables distantes mentionnés, soit de manière directe, soit comme parties de listes d'interfaces et de signaux, dans les listes de signaux de sortie associées aux portes du type d'agent."

4.25 Clarification – Modèle de type d'agent – Paragraphe 8.1.1.1: Types d'agent, Modèle

La description de la transformation du modèle était incomplète.

Remplacer la totalité du paragraphe *Modèle* par le texte suivant:

"Un agent possédant un corps <agent body> ou une zone <agent body area> constitue une notation abrégée pour un type d'agent possédant uniquement une machine d'état, mais ne contenant pas d'agents. La machine d'état est obtenue en remplaçant le corps <agent body> ou la zone <agent body area> par une définition d'état composite. Cette définition d'état composite a le même nom que le type d'agent et son graphe *State-transition-graph* est représenté par le corps <agent body> ou par la zone <agent body area>.

Un type d'agent possédant:

- une partition <state partition> avec une référence <composite state reference> ou un état <composite state>, ou
- une partition <state partition area> avec une zone <composite state reference area>, ou
- une zone <composite state area>

constitue une notation abrégée pour un type d'agent possédant une machine d'état basée sur un type d'état composite impliqué virtuel. Le type d'état impliqué possède le corps de la référence <composite state reference>, de l'état <composite state>, de la zone <composite state reference area> ou de la zone <composite state area>. Si le type d'agent est un sous-type et si le supertype possède une partition <state partition> ou une zone <state partition area>, le type d'état impliqué est un sous-type qui hérite de manière implicite du type d'état impliqué par le supertype.

Chaque type d'état impliqué possède une contrainte constituée du type lui-même (se référer au paragraphe 8.3.1)."

4.26 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 8.1.1.2: Type de système, Grammaire textuelle concrète

Modifier comme suit la règle <system type reference>:

```
<system type reference> ::=  
    system type <type reference heading> <type reference properties>
```

et la faire suivre du nouvel alinéa:

"Un en-tête <type reference heading> qui fait partie d'une règle <system type reference> doit posséder un nom <system type name>."

4.27 Erreur d'écriture – Syntaxe simplifiée – Paragraphe 8.1.1.2: Type de système, Grammaire textuelle concrète

Simplifier la syntaxe en remplaçant la structure "<agent type structure>" par la structure "<agent structure>" dans la règle <system type definition>.

```
<system type definition> ::=  
    <package use clause> *  
    <system type heading> <end> <agent structure>  
    endsystem type [ [<qualifier>] <system type name>] <end>
```

4.28 Erreur d'écriture – Accès omis/syntaxe simplifiée – Paragraphe 8.1.1.2: Type de système, Grammaire graphique concrète

Dans la règle <system type diagram> remplacer "<agent type diagram content>" par "<agent type structure area>". La simplification de syntaxe ci dessous remplace toutefois "<agent type diagram content>" par "<agent structure area>".

Se référer également au paragraphe 9.1.

La syntaxe est simplifiée de manière à utiliser la zone <agent structure area>. Remplacer la règle "<system diagram>" par:

```
<system type diagram> ::=  
    <frame symbol> contains {<system type heading> <agent structure area> }  
    is connected to { { <gate on diagram> * } set }  
    [ is associated with <package use area> ]
```

4.29 Clarification – Paragraphe 8.1.1.2: Type de système, Grammaire graphique concrète

Remplacer le texte "<graphical type reference heading> qui contient" par le texte "<type reference heading> avec" de manière à obtenir l'alinéa révisé suivant:

"La zone <type reference area> qui fait partie d'une zone <system type reference area> doit avoir un en-tête <type reference heading> avec un nom <system type name>".

4.30 Clarification – Simplification de la syntaxe – Paragraphe 8.1.1.3: Type de bloc, Grammaire textuelle concrète

Modifier la règle "<block type definition>" de manière à utiliser la structure "<agent structure>" à la place de la structure "<agent type structure>".

4.31 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 8.1.1.3: Type de bloc, Grammaire textuelle concrète

Modifier comme suit la règle <block type reference>:

<block type reference> ::=
 <type preamble>
 block type <type reference heading> <type reference properties>

en la faisant suivre du nouvel alinéa:

"Un en-tête <type reference heading> qui fait partie d'une règle <block type reference> doit posséder un nom <block type name>."

4.32 Clarification – Paragraphe 8.1.1.3: Type de bloc, Grammaire graphique concrète

Dans la règle <block type diagram> remplacer "<agent type diagram content>" par "<agent type structure area>". Remplacer toutefois ensuite "<agent type structure area>" par "<agent structure area>" pour simplifier la syntaxe.

4.33 Clarification – Paragraphe 8.1.1.3: Type de bloc, Grammaire graphique concrète

Remplacer le texte "<graphical type reference heading> contenant" par le texte "<type reference heading> avec" de manière à obtenir le texte suivant:

"La zone <type reference area> qui fait partie d'une zone <block type reference area> doit avoir un en-tête <type reference heading> avec un nom <block type name>."

4.34 Correction d'erreur – Paragraphe 8.1.1.3: Type de bloc, Grammaire graphique concrète

Supprimer la phrase:

"Le diagramme <gate on diagram> dans un diagramme <block type diagram> peut ne pas inclure d'identificateurs <external channel identifiers>."

4.35 Clarification – Simplification de la syntaxe – Paragraphe 8.1.1.4: Type de processus, Grammaire textuelle concrète

Modifier la règle <process type definition> de manière à utiliser la structure "<agent structure>" à la place de la structure "<agent type structure>".

4.36 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 8.1.1.4: Type de processus, Grammaire textuelle concrète

Modifier comme suit la règle <process type reference>:

<process type reference> ::=
 <type preamble>
 process type <type reference heading> <type reference properties>

en la faisant suivre du nouvel alinéa:

"Un en-tête <type reference heading> qui fait partie d'une règle <process type reference> doit posséder un nom <process type name>."

4.37 Clarification – Paragraphe 8.1.1.4: Type de processus, Grammaire graphique concrète

Dans la règle <process type diagram> remplacer "<agent type diagram content>" par "<agent type structure area>". Remplacer toutefois ensuite "<agent type structure area>" par "<agent structure area>" pour simplifier la syntaxe.

4.38 Clarification – Paragraphe 8.1.1.4: Type de processus, Grammaire graphique concrète
Remplacer le texte "<graphical type reference heading> qui contient" par le texte "<type reference heading> avec" de manière à obtenir l'alinéa révisé suivant:

"La zone <type reference area> qui fait partie d'une zone <process type reference area> doit avoir un en-tête <type reference heading> avec un nom <process type name>."

4.39 Correction d'erreur – Paragraphe 8.1.1.4: Type de processus, Sémantique

Supprimer la phrase:

"Le diagramme <gate on diagram> dans un diagramme <process type diagram> peut ne pas inclure d'identificateurs <external channel identifiers>."

4.40 Erreur d'écriture – Aligner avec l'Annexe F – Paragraphe 8.1.1.5: Type d'état composite, Grammaire abstraite

Supprimer "*Connect-node-set*" dans la définition *Composite-state-type-definition* de manière à obtenir la règle:

```
Composite-state-type-definition  ::  State-type-name
                                   [ Composite-state-type-identifieur ]
                                   Composite-state-formal-parameter*
                                   State-entry-point-definition-set
                                   State-exit-point-definition-set
                                   Gate-definition-set
                                   Data-type-definition-set
                                   Syntype-definition-set
                                   Exception-definition-set
                                   Composite-state-type-definition-set
                                   Variable-definition-set
                                   Procedure-definition-set
                                   [ Composite-state-graph | State-aggregation-node ]
```

Remplacer "::" par "=" dans *Composite-state-type-identifieur* de manière à obtenir la règle:

```
Composite-state-type-identifieur  =  Identifieur
```

4.41 Correction d'erreur – Paragraphe 8.1.1.5: Type d'état composite, Grammaire abstraite

Il n'est pas dangereux qu'un type d'état composite possède des accès non utilisés; la condition statique suivante est supprimée en conséquence:

"L'ensemble *Gate-definition-set* doit être vide sauf si l'état composite est utilisé en tant que définition *State-machine-definition*."

4.42 Clarification – Paragraphe 8.1.1.5: Type d'état composite, Grammaire textuelle concrète

Modifier comme suit la règle <composite state type definition> par:

```
<composite state type definition> ::=
    {<package use clause>}*
    { <composite state type heading> | <state aggregation type heading> }<end>
    <composite state structure>
    endsubstructure state type [ [ <qualifier> ] <composite state type name> ] <end>
```

4.43 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 8.1.1.5: Type d'état composite, Grammaire textuelle concrète

Modifier comme suit la règle <composite state type reference> par:

```
<composite state type reference> ::=  
    <type preamble>  
    state type <type reference heading> <type reference properties>
```

en la faisant suivre du nouvel alinéa:

"Un en-tête <type reference heading> qui fait partie d'une règle <composite state type reference> doit posséder un nom <composite state type name>."

4.44 Clarification – Paragraphe 8.1.1.5: Type d'état composite, Grammaire textuelle concrète

Modifier comme suit la règle <composite state type diagram>:

```
<composite state type diagram> ::=  
    <frame symbol>  
    contains {  
        { <composite state type heading> | <state aggregation type heading> }  
        <composite state structure area>  
        is associated with { <graphical state connection point>* } set  
        is connected to { { <gate on diagram>* } set }  
        [ is associated with <package use area> ]
```

4.45 Clarification – Paragraphe 8.1.1.5: Type d'état composite, Grammaire graphique concrète

Remplacer le texte "<graphical type reference heading> qui contient" par le texte "<type reference heading> avec" de manière à obtenir l'alinéa révisé suivant:

La zone <type reference area> qui fait partie d'une règle <composite state type reference area> doit avoir un en-tête <type reference heading> avec un nom <composite state type name>.

4.46 Correction d'erreur – Paragraphe 8.1.1.5: Type d'état composite, Grammaire graphique concrète

Déplacer l'alinéa:

"Le diagramme <gate on diagram> dans un diagramme <process type diagram> doit être à l'extérieur du cadre du diagramme."

du paragraphe *Sémantique* vers la fin du paragraphe *Grammaire graphique concrète*.

Supprimer la phrase suivante dans le paragraphe *Sémantique* (qui devrait appartenir à la *Grammaire graphique concrète*):

"Le diagramme <gate on diagram> dans un diagramme <composite state type diagram> peut ne pas inclure d'identificateurs <external channel identifiers>."

4.47 Erreur d'écriture – Type souligné – Paragraphe 8.1.3.1: Définition de système fondée sur le type de système, Grammaire textuelle concrète

Remplacer "<system type expression>" par "<system type expression>" dans la règle <typebased system heading>.

4.48 Erreur d'écriture – Transformation non nécessaire – Paragraphe 8.1.3.1: Définition de système fondée sur le type de système

Remplacer:

"Sémantique

Une définition de système fondée sur le type définit une définition *Agent-definition* de *Agent-kind* **SYSTEM** obtenue par transformation à partir d'un type de système.

Modèle

Une définition <textual typebased system definition> ou <graphical typebased system definition> est transformée en une définition <system definition> qui possède les définitions du type de système telles qu'elles sont définies par l'expression <system type expression>."

par:

"Sémantique

Une définition de système fondée sur le type indiqué par une définition <textual typebased system definition> ou une définition <graphical typebased system definition> constitue une définition *Agent-definition* pour le système *Agent-kind* **SYSTEM** qui est une instantiation du type de système indiqué par l'expression <system type expression>."

4.49 Erreur d'écriture – Transformation non nécessaire – Paragraphe 8.1.3.2: Définition de bloc fondée sur le type de bloc

Remplacer:

"Sémantique

Une définition de bloc fondée sur le type définit des définitions *Agent-definitions* de *Agent-kind* **BLOCK** obtenues par transformation à partir d'un type de bloc.

Modèle

Une définition <textual typebased block definition> ou <graphical typebased block definition> est transformée en une définition <block definition> qui possède les définitions du type de bloc telles qu'elles sont définies par l'expression <block type expression>.

Une définition <inherited block definition> peut seulement apparaître dans une définition de sous-type. Elle représente le bloc défini dans le supertype de la définition du sous-type. Il est possible de spécifier des canaux supplémentaires connectés aux accès du bloc hérité."

par:

"Une définition <inherited block definition> peut figurer uniquement dans une définition de sous-type. Elle représente le bloc défini dans le supertype de la définition du sous-type."

NOTE – Il est permis de spécifier des canaux supplémentaires connectés à des accès d'un bloc hérité.

Sémantique

Une définition de bloc fondée sur le type indiqué par une définition <textual typebased block definition> ou <graphical typebased block definition> constitue une définition *Agent-definition* pour un bloc *Agent-kind* **BLOCK** qui est une instantiation du type de bloc indiqué par l'expression <block type expression>."

4.50 Erreur d'écriture – Transformation non nécessaire – Paragraphe 8.1.3.3: Définition de processus fondée sur le type de processus

Remplacer:

"Sémantique

Une définition de processus fondée sur le type définit une définition *Agent-définition* avec *Agent-kind* **PROCESS** obtenue par transformation à partir d'un type de processus.

Modèle

Une définition <textual typebased process definition> ou <graphical typebased process definition> est transformée en une définition <process definition> qui possède les définitions du type de processus telles qu'elles sont définies par l'expression <process type expression>.

Une définition <inherited process definition> peut seulement apparaître dans une définition de sous-type. Elle représente le processus défini dans le supertype de la définition du sous-type. Il est possible de spécifier des canaux supplémentaires connectés aux accès du processus hérité."

par:

"Une définition <inherited process definition> peut figurer uniquement dans une définition de sous-type. Elle représente le processus défini dans le supertype de la définition du sous-type.

NOTE – Il est permis de spécifier des canaux supplémentaires connectés à des accès d'un bloc hérité."

Sémantique

Une définition de processus fondée sur le type indiqué par une définition <textual typebased process definition> ou <graphical typebased process definition> constitue une définition *Agent-définition* pour un processus *Agent-kind* **PROCESS** qui est une instanciation du type de processus indiqué par l'expression <process type expression>."

4.51 Erreur d'écriture – Export absent – Paragraphe 8.1.4: Type abstrait, Grammaire concrète

Supprimer les deux alinéas:

"Un identificateur <operation identifier> ne doit pas représenter l'identificateur *Operation-identifier* pour l'opérateur Make défini pour un type de données abstrait.

Un identificateur <literal identifier> ne doit pas représenter un identificateur *Literal-identifier* désignant une signature *Literal-signature* d'un type de données abstrait."

4.52 Erreur d'écriture – Sorte dans une propriété de paramètre de signal – Paragraphe 8.1.5: Références de type, Grammaire textuelle concrète

Modifier comme suit la production pour la règle <signal parameter property>:

<signal parameter property> ::=
 <sort>

et remplacer l'alinéa:

"Une propriété <signal parameter property> correspond à une liste de paramètres de signal dans une définition de signal. La liste <sort list> doit correspondre à la liste <sort list> dans la définition <signal definition item> de la définition de signal correspondante. Il ne doit y avoir qu'une seule propriété <signal parameter property> dans les propriétés <type reference properties> ou dans la zone <type reference area>."

par:

"Une propriété <signal parameter property> correspond à un paramètre de signal dans une définition de signal. L'élément sorte doit correspondre à un élément <sort> de la liste <sort list> dans la définition <signal definition item> de la définition de signal correspondante. Les propriétés <signal parameter property> dans les propriétés <type reference properties> doivent être présentes dans le même ordre que les propriétés correspondantes dans la définition de type de référence."

4.53 Erreur d'écriture – Export absent – Paragraphe 8.1.5: Références de type, Grammaire graphique concrète

Ceci était autorisé dans le langage SDL-92 et a été omis de manière erronée dans le langage SDL-2000.

Ajouter "**exported as**" dans la règle <type reference heading>:

```
<type reference heading> ::=  
    <type preamble>  
    [ exported [ as <remote procedure identifier> ] ]  
    [<qualifier>] <name> [<formal context parameters>]
```

Compléter l'alinéa qui fait suite à cette règle par le texte suivant:

"Si le mot clé **exported** figure dans une règle <type reference heading>, le type référencé doit être une procédure exportée et si un identificateur <remote procedure identifier> est également présent, la procédure doit alors identifier la même définition de procédure distante."

4.54 Erreur d'écriture – Contrainte de point d'extrémité trop restrictive – Paragraphe 8.1.6: Accès, Grammaire textuelle concrète

L'alinéa suivant est modifié parce qu'il contient une restriction devenue inutile:

"L'identificateur <identifier> de la contrainte <textual endpoint constraint> doit indiquer une définition de type de bloc, de type de processus ou de type d'état."

4.55 Erreur d'écriture – Identificateur erroné – Paragraphe 8.2.1: Paramètre de contexte de type d'agent, Grammaire textuelle concrète

La contrainte doit être un identificateur <agent type identifier> et non un identificateur <agent identifier>.

4.56 Clarification – Signification du double point – Paragraphe 8.2.2: Paramètre de contexte d'agent, Grammaire textuelle concrète

Ajouter le texte "(<colon> <agent type identifier>)" à l'alinéa:

"Un paramètre d'agent réel doit identifier une définition d'agent. Son type doit être un sous-type du type de processus de contrainte (**atleast** <agent type identifier>) sans adjonction de paramètres formels aux paramètres du type de contrainte, ou il doit être le type désigné par l'identificateur <agent type identifier>(<colon> <agent type identifier>), ou il doit être compatible avec la signature formelle <agent signature>."

4.57 Clarification – Lien vers la syntaxe abstraite – Paragraphe 8.3.1: Adjonction de propriétés, Grammaire textuelle concrète

Ajouter le texte suivant avant l'alinéa débutant par "La syntaxe concrète pour la spécialisation de types de données":

"L'expression <type expression> de la règle <specialization> représente:

- a) dans l'en-tête <agent additional heading>, l'identificateur *Agent-type-identifier* de la définition *Agent-type-definition* du paragraphe 8.1.1.1;

- b) dans l'en-tête <composite state type heading> ou <state aggregation type heading>, l'identificateur *Composite-state-type-identifieur* de la définition *Composite-state-type-definition* du paragraphe 8.1.1.5;
- c) dans l'en-tête <procedure heading>, l'identificateur *Procedure-identifieur* de la définition *Procedure-definition* du paragraphe 9.4."

4.58 Erreur d'écriture – Référence – Paragraphe 8.3.3: Transition/Sauvegarde virtuelle, Grammaire textuelle concrète

Remplacer la référence "11.2.2 (entrée virtuelle)" par "11.3 (entrée virtuelle)".

4.59 Erreur d'écriture – Référence à l'Annexe F – Paragraphe 8.3.3: Sémantique

Remplacer l'alinéa:

"La transformation de transitions et sauvegardes virtuelles dans les états astérisques est élaborée dans l'Annexe F."

par:

"La définition faite dans le sous-type définit la transition virtuelle ou la sauvegarde. Une transition virtuelle ou une sauvegarde qui n'est pas redéfinie dans une définition de sous-type hérite de la définition faite pour le supertype."

4.60 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 8.4: Associations, Grammaire textuelle concrète

Modifier comme suit la règle <association end>:

```
<association end> ::=
    [<visibility>] [as <role name>] [size <multiplicity>] [ordered]
    { <agent type reference> | <interface reference> | <data type reference> }
```

Supprimer la règle <linked type identifier> qui est inutile.

Supprimer le texte "<linked type identifier> dans" dans l'alinéa débutant par "Si une extrémité <association end> est précédée ...".

4.61 Clarification – Paragraphe 9: Agents, Grammaire abstraite

La règle de syntaxe abstraite pour le graphe *State-transition-graph* est supprimée et déplacée vers le paragraphe 11.11.1.

Les règles de syntaxe abstraite pour *State-machine-definition*, *Agent-kind* et *Agent-formal-parameter* sont supprimées et déplacées vers le paragraphe 8.1.1.1.

4.62 Clarification – Paragraphe 9: Agents, Grammaire textuelle concrète

Remplacer "<state partitioning>" par "<state partition>" dans la règle <agent structure>.

Remplacer les deux occurrences de "<state partitioning>" par "<state partition>" dans l'alinéa qui fait suite aux règles.

4.63 Erreurs administratives – Paragraphe 9: Agents, Grammaire textuelle concrète

Ajouter les conditions manquantes sur les connexions. Insérer le nouvel alinéa suivant avant la règle <agent instantiation>:

"Une connexion <channel to channel connection> ne doit pas être contenue dans une définition <agent type definition>."

Insérer la variante "<composite state>" qui a été omise par erreur dans la règle <entity in agent>.

4.64 Clarification – Paragraphe 9: Agents, Grammaire textuelle concrète

Supprimer la règle <state machine graph> en supprimant les lignes:

<state machine graph>

<state machine graph>::=

4.65 Clarification – Simplification – Paragraphe 9: Agents, Grammaire textuelle concrète

Afin de simplifier la syntaxe de la règle <agent body> permettant son utilisation à la fois pour les types d'agent et les agents, il convient de rendre optionnelle l'expression [<on exception>] <start>:

[[<on exception>] <start>]

Insérer le nouvel alinéa suivant avant l'alinéa "L'utilisation et la syntaxe de l'ensemble <valid input signal set> sont définis au paragraphe 9.":

"Le départ <start> peut être omis dans la règle <agent body> uniquement dans les définitions de type d'agent."

4.66 Clarification – Mode de description – Paragraphe 9: Agents, Grammaire graphique concrète

Remplacer le nom de la règle "<agent diagram content>" par "<agent structure area>" et remplacer dans cette règle la zone "<type in agent area>" par le diagramme "<entity in agent diagram>".

Redéfinir comme suit la règle <agent body area>:

<agent body area> ::=

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

Ceci incorpore la règle <state machine graph area> qui est supprimée de ce fait.

Remplacer le nom de la règle "<type in agent area>" (supprimée dans le paragraphe 8.1.1.1 Types d'agent) par "<entity in agent diagram>" et l'insérer sous la forme suivante avant la règle <interaction area>:

<entity in agent diagram> ::=

| <agent type diagram>
| <agent type reference area>
| <composite state area>
| <composite state type diagram>
| <composite state type reference area>
| <procedure diagram>
| <procedure reference area>
| <data type reference area>
| <signal reference area>
| <association area>

Dans la règle <interaction area>, remplacer "<state machine area>" par "<state partition area>".

Supprimer la règle <state machine area>.

Dans la règle <create line endpoint area>, remplacer "<state machine area>" par "<state partition area>".

Modifier comme suit la première phrase faisant suite aux règles:

"Une règle <agent text area> peut contenir une référence <agent reference> seulement si la zone <agent structure area> qui l'englobe directement contient une zone <interaction area>."

Remplacer "<state machine area>" par "<state partition area>" dans la phrase qui suit la NOTE 1 de manière à obtenir la phrase:

"La zone <state partition area> de la zone <interaction area> désigne l'automate à états (composites) de l'agent, qui peut être représenté directement sous la forme d'un graphe d'agent ou par référence à une définition d'état."

Insérer à la suite du paragraphe le nouvel alinéa suivant:

"La zone <start area> peut uniquement être omise dans un diagramme de type d'agent."

4.67 Erreur d'écriture – Mode de description – Paragraphe 9: Agents, Grammaire graphique concrète

Modifier comme suit l'alinéa débutant par "S'il existe une zone <agent reference area> ...":

"S'il existe une zone <agent reference area> pour l'agent, la ou les zones <gate property area> associées à la zone <agent reference area> correspondent aux accès <gate on diagram> associés aux diagrammes <agent diagram>. Aucune zone <gate property area> associée à la zone <agent reference area> ne doit contenir des éléments <signal list item> qui ne sont pas contenus dans les accès <gate on diagram> associés au diagramme <agent diagram>. Une règle correspondante s'applique s'il existe une référence <agent reference> pour l'agent.

4.68 Erreur d'écriture – Restriction non nécessaire pour une connexion d'accès – Paragraphe 9: Agents, Sémantique

L'alinéa suivant n'est pas correct en raison de la possibilité de canaux implicites et il est supprimé de ce fait. Le texte original SDL-92 était correct.

"Si une définition <agent definition> ou <agent type definition>, utilisée dans une définition <textual typebased agent definition>, contient une ou plusieurs définitions <channel definition> et une ou plusieurs définitions <textual typebased agent definition>, alors chaque accès des définitions <agent type definition> des définitions <textual typebased agent definition> contenues doit être connecté à un canal.

4.69 Clarification – Conditions d'arrêt pour des agents initiaux – Paragraphe 9: Agents, Sémantique

Afin de clarifier la durée de vie d'un agent qui ne possède pas de machine d'état, insérer l'alinéa suivant à la suite de l'alinéa traitant des conditions d'arrêt qui débute par "Lorsque la machine à états d'un agent ..."

"Si un agent ne possède pas de machine d'état explicite ou implicite, il passe alors dans une condition d'arrêt dès la création des agents contenus initialement. Un agent qui contient des instances initiales et aucune machine d'état cesse, de ce fait, d'exister immédiatement après sa création".

4.70 Erreur d'écriture – Ensemble de signaux d'entrée complets valides – Paragraphe 9: Agents, Sémantique

Le texte

- "a) l'ensemble des signaux se trouvant dans tous les canaux ou accès conduisant à l'ensemble des instances d'agent;
- b) l'ensemble <valid input signal set>;
- c) les signaux d'entrée implicites définis en 10.5, 10.6, 11.4, 11.5 et 11.6;
- d) "

est remplacé par le texte suivant:

- "a) l'ensemble des signaux se trouvant dans tous les canaux ou accès conduisant à la machine d'état de l'agent;
- b) l'ensemble <valid input signal set> de l'agent;
- c) l'ensemble <valid input signal set> de la machine d'état de l'agent;
- d) les signaux d'entrée implicites définis dans les paragraphes 10.5 et 10.6;
- e) "

4.71 Erreur d'écriture – Pluriel incorrect – Paragraphe 9: Agents, Sémantique

[Cetle erreur n'affecte pas la version française]

4.72 Erreur d'écriture – Définition de l'ensemble de sortie complet – Paragraphe 9: Agents, Sémantique

Insérer le texte suivant à la fin du paragraphe *Sémantique*:

"L'ensemble de sortie complet d'un ensemble d'agents est le même que l'ensemble de sortie complet du type de l'ensemble d'agents."

4.73 Clarification – Valeur pid des agents contenus – Paragraphe 9: Agents, Modèle

Généraliser, en le remplaçant par le texte suivant, l'avant-dernier alinéa du paragraphe *Modèle* de manière à traiter des instances initiales pour tout agent créé (en plus du système):

"L'élément **parent** est initialisé avec la valeur "Nulle" pour toutes les instances d'agent qui sont créées au moment de la création de l'instance qui les contient."

4.74 Erreur d'écriture – Accès omis – Paragraphe 9.1: Système, Grammaire graphique concrète

Modifier comme suit la règle <system diagram>:

```
<system diagram> ::=
    <frame symbol> contains {<system heading> <agent structure area> }
    is connected to { {<gate on diagram>}* <external channel identifiers> } set
    [ is associated with <package use area> ]
```

4.75 Erreur d'écriture – Condition incorrecte – Paragraphe 9.1: Système, Grammaire graphique concrète

Modifier comme suit la dernière phrase:

"Une zone <system reference area> doit être utilisée uniquement comme partie d'une zone <specification area>."

4.76 Clarification – Paragraphe 9.2: Bloc, Grammaire graphique concrète

Dans la règle <block diagram>, remplacer "<agent diagram content>" par "<agent structure area>" et modifier:

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

de manière à obtenir

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

4.77 Clarification – Paragraphe 9.3: Processus, Grammaire graphique concrète

Dans la règle <process diagram>, remplacer "<agent diagram content>" par "<agent structure area>" et modifier:

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

de manière à obtenir:

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

4.78 Erreur d'écriture – Nœud de départ/Identificateur de procédure de base – Paragraphe 9.4: Procédure, Grammaire abstraite

Dans la définition de *Procedure-definition*, remplacer "*Procedure-identifier*" par "[*Procedure-identifier*]".

Dans la définition de *Procedure-graph*, remplacer "*Procedure-start-node*" par "[*Procedure-start-node*]".

Ajouter l'alinéa suivant à la fin de la *Grammaire abstraite*:

"Dans une spécification *SDL-specification*, toutes les procédures potentiellement instanciables doivent posséder un nœud *Procedure-start-node*."

4.79 Erreur d'écriture – Export absent – Paragraphe 9.4: Procédure, Grammaire textuelle concrète

Ceci était autorisé dans le langage SDL-92 et a été omis par erreur dans le langage SDL-2000.

Modifier comme suit la règle <procedure reference>:

<procedure reference> ::=
 <type preamble> [**exported** [**as** <remote procedure identifier>]]
 procedure <procedure identifier> <type reference properties>

Ajouter le nouvel alinéa suivant avant la *Grammaire graphique complète*:

"Si le mot clé **exported** figure dans une référence de procédure, la procédure faisant l'objet de la référence doit alors être une procédure exportée et si un identificateur <remote procedure identifier> est également présent, la procédure doit alors identifier la même définition de procédure distante."

4.80 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 9.4: Procédure, Grammaire textuelle concrète

Modifier comme suit la règle <procedure reference>:

<procedure reference> ::=
 <type preamble>
 procedure <type reference heading> <type reference properties>

et la faire suivre du nouvel alinéa:

"Un en-tête <type reference heading> qui fait partie d'une règle <procedure reference> doit posséder un nom <procedure name>."

4.81 Clarification – Paragraphe 9.4: Procédure, Grammaire graphique concrète

Dans la règle <procedure diagram> remplacer "<procedure graph area>" par "<procedure body area>" parce que la règle a été renommée (voir ci-dessous).

4.82 Erreur d'écriture – Etat composite absent – Paragraphe 9.4: Procédure, Grammaire graphique concrète

Modifier comme suit la règle `<procedure area>` de manière à autoriser des diagrammes et des références avec un type d'état composite:

```
<procedure area> ::=
    | <procedure diagram>
    | <procedure reference area>
    | <composite state type diagram>
    | <composite state type reference area>
```

4.83 Clarification – Paragraphe 9.4: Procédure, Grammaire graphique concrète

Remplacer le texte "<graphical type reference heading> qui contient" par le texte "<type reference heading> avec" de manière à obtenir l'alinéa révisé suivant:

"La zone <type reference area> qui fait partie d'une zone <procedure reference area> doit avoir un en-tête <type reference heading> avec un nom <procedure name>."

4.84 Clarification – Paragraphe 9.4: Procédure, Grammaire graphique concrète

Remplacer le nom de la règle "<procedure graph area>" par "<procedure body area>".

Remplacer "<procedure graph area>" par "<procedure body area>" pour obtenir l'alinéa révisé suivant:

"La zone <on exception association area> d'une zone <procedure body area> identifie le gestionnaire d'exception associé au graphe complet. L'extrémité de départ ne doit être reliée à aucun symbole."

4.85 Erreur d'écriture – Paragraphe 9.4: Procédure, Sémantique

Dans la phrase:

"a) une variable locale est créée pour tout paramètre *In-parameter* ayant le *Name* et la *Sort* du paramètre *In-parameter*...",

Remplacer le trait d'union du dernier "*In-parameter*" par un trait d'union normal.

[*Cette erreur n'affecte pas la version française*]

4.86 Erreur d'écriture – Référence à l'Annexe F – Paragraphe 9.4: Procédure, Sémantique

Remplacer l'alinéa:

"Une procédure externe est une procédure dont le corps <procedure body> n'est pas inclus dans la description en SDL. De manière conceptuelle, il est supposé qu'un corps <procedure body> est donné à une procédure externe et que celle-ci sera transformée en une définition <procedure definition> qui fait partie de la transformation du système générique (voir Annexe F)."

par:

"Une procédure externe est une procédure dont le corps <procedure body> ne fait pas partie de la description SDL (se référer au paragraphe 13)."

4.87 Erreur d'écriture – Référence à l'Annexe F – Paragraphe 10.1: Canal, Grammaire textuelle concrète

Remplacer l'alinéa:

"Si l'accès <gate> est spécifié, le canal est relié à cet accès. L'accès et le canal doivent avoir au minimum un élément en commun dans leurs listes de signaux dans la même direction. Si aucun accès <gate> n'est spécifié, les règles suivantes s'appliquent:

- a) si l'extrémité de canal est un agent ou une machine à états et que cet agent/état contient une connexion <channel to channel connection> pour le canal, ce dernier est relié à l'accès implicite introduit par la connexion <channel to channel connection>,
- b) si l'extrémité de canal est un état, le canal est relié à l'accès implicite de cette machine à états (voir Annexe F),

autrement, le canal introduit un accès implicite sur l'agent ou l'état mentionné dans l'extrémité <channel endpoint>. Cet accès obtient la liste <signal list> des trajets <channel path> respectifs comme ses contraintes d'accès correspondantes. Le canal est relié à cet accès."

par:

"Si l'accès <gate> est spécifié, le canal est alors connecté à cet accès. L'accès et le canal doivent posséder au moins un élément commun dans leurs listes de signaux dans la même direction. Les règles suivantes s'appliquent si aucun accès <gate> n'est spécifié. Si l'extrémité de canal est un agent ou une machine d'état qui contient une connexion <channel to channel connection> pour le canal, ce dernier est alors connecté à l'accès implicite introduit par la connexion <channel to channel connection>. Sinon le canal introduit un accès implicite sur l'agent ou l'état mentionné dans l'extrémité <channel endpoint>. Cet accès obtient la liste <signal list> des trajets <channel path> respectifs comme ses contraintes d'accès correspondantes. Le canal est connecté à cet accès."

Ajouter l'alinéa suivant à la fin du paragraphe:

"Les règles suivantes doivent être satisfaites si une règle <channel definition> contient deux trajets <channel path>:

- le point d'extrémité <channel endpoint> qui fait suite au mot clé **from** dans la première règle <channel path> doit être le même que le point d'extrémité <channel endpoint> qui fait suite au mot clé **to** dans la deuxième règle <channel path>;
- le point d'extrémité <channel endpoint> qui fait suite au mot clé **to** dans la première règle <channel path> doit être le même que le point d'extrémité <channel endpoint> qui fait suite au mot clé **from** dans la deuxième règle <channel path>."

4.88 Clarification – Mode de description – Paragraphe 10.1: Canal, Grammaire graphique concrète

Remplacer l'alinéa débutant par "Pour chaque flèche placée sur le symbole <channel symbol>" par le texte suivant:

"Il doit y avoir une zone <signal list area> au plus pour chaque flèche placée sur le symbole <channel symbol>. Chaque zone <signal list area> doit être suffisamment proche de la flèche à laquelle elle est associée pour éviter toute ambiguïté. Cette flèche indique la direction de la liste de signaux de trajet de canal à laquelle elle est associée."

4.89 Clarification – Mode de description – Paragraphe 10.1: Canal, Grammaire graphique concrète

Dans la règle <channel definition area> remplacer les deux instances de "<state machine area>" par "<state partition area>" et remplacer comme suit la phrase qui fait suite aux règles syntactiques:

"Lorsqu'un symbole <channel symbol> est connecté à une zone <state partition area>, il désigne la machine d'état de l'agent englobant directement la définition de canal."

4.90 Erreur d'écriture – Référence à l'Annexe F – Paragraphe 10.1: Canal, Modèle

Le texte:

"De plus amples détails sont présentés dans l'Annexe F."

avait été remplacé par:

"De plus amples détails sont donnés dans le modèle pour des canaux implicites."

Le modèle a toutefois été déplacé par la suite depuis le projet d'Annexe F vers le paragraphe 10.1 et le texte du modèle de l'Annexe F a été révisé pour constituer le texte suivant du paragraphe *Modèle*:

"Modèle

Si le nom <channel name> ne figure pas dans une définition <channel definition> ou une zone <channel definition area>, le canal est alors nommé de manière implicite et non ambiguë.

Un canal dont les deux extrémités constituent les accès d'une même définition <textual typebased agent definition> représente un canal individuel de chacun des agents de cet ensemble vers tous les agents de l'ensemble, y compris l'agent d'origine. Tout canal bidirectionnel résultant qui connecte un agent de l'ensemble à cet agent lui-même est subdivisé en deux canaux unidirectionnels.

Si un agent ou un type d'agent contient des accès explicites ou implicites qui ne sont pas connectés par des canaux implicites, des canaux explicites sont alors déduits conformément aux trois transformations suivantes qui s'appliquent après l'application de la transformation de création basée sur le type défini dans le paragraphe 11.13.2.

- 1) *Transformation 1*: insertion de canaux entre des ensembles d'instances au sein de l'agent ou du type d'agent et entre les ensembles d'instances et la machine d'état de l'agent.
- 2) *Transformation 2*: insertion de canaux entre un accès sur l'agent ou le type d'agent et des ensembles d'instance au sein de l'agent, ou entre le type d'agent et des accès sur la machine d'état de l'agent.
- 3) *Transformation 3*: insertion de canaux entre des accès sur des ensembles d'instances au sein de l'agent ou du type d'agent ou des accès sur la machine d'état de l'agent d'une part, vers des accès sur l'agent ou le type d'agent d'autre part.

Ces transformations sont décrites ci-dessous de manière détaillée. Elles s'appliquent dans l'ordre indiqué.

Dans les transformations, un élément d'une liste de signaux (interfaces, signaux, procédures distantes ou variables distantes) correspond à un autre élément d'une liste de signaux si l'une des conditions suivantes est satisfaite:

- a) les deux éléments concernent les mêmes interfaces, signaux, procédures distantes ou variables distantes;
- b) le premier élément concerne un signal, une procédure distante ou une variable distante et le deuxième une interface qui inclut le signal, la procédure distante ou la variable distante;
- c) les deux éléments concernent des interfaces et l'élément de la deuxième liste de signaux hérite de l'élément de la première liste de signaux.

Transformation 1: insertion de canaux implicites entre entités au sein d'un agent ou d'un type d'agent

- a) si un élément de la liste de signaux de sortie associée à un accès d'une instance au sein d'un agent (ou d'un type d'agent) correspond à un élément d'une liste de signaux en entrée associée à un accès d'une autre instance au sein du même agent (ou, respectivement, du type d'agent); et
- b) si aucun de ces accès ne possède de canal connecté

alors:

- i) si aucun canal implicite n'existe entre les deux accès, un canal implicite unidirectionnel est alors créé entre l'accès de l'élément en entrée et l'accès de l'élément de sortie; ce canal est sans retard s'il se trouve au sein d'un processus (ou d'un type de processus), il est avec retard dans le cas contraire; et
- ii) l'élément est ajouté à la liste de signaux du canal implicite.

Transformation 2: insertion de canaux implicites issus d'un accès sur un agent ou un type d'agent

- a) si un élément de la liste de signaux en entrée associée à un accès en-dehors d'un agent (ou d'un type d'agent) correspond à un élément d'une liste de signaux en entrée associée à un accès d'une instance au sein de l'agent (ou, respectivement, du type d'agent); et
- b) s'il n'existe pas de canal explicite au sein de l'agent (ou, respectivement, du type d'agent) connecté à l'accès en dehors de l'agent (ou, respectivement, du type d'agent) ni de canal explicite connecté à l'accès de l'instance au sein de l'agent (ou, respectivement, du type d'agent)

alors:

- i) si aucun canal implicite n'existe entre les deux accès, un canal implicite unidirectionnel est alors créé depuis l'accès en dehors de l'agent (ou, respectivement, du type d'agent) vers l'accès de l'instance au sein de l'agent (ou, respectivement, du type d'agent); ce canal est sans retard s'il se trouve au sein d'un processus (ou d'un type de processus), il est avec retard dans le cas contraire; et
- ii) l'élément est ajouté à la liste de signaux du canal implicite.

Transformation 3: insertion de canaux implicites issus d'un accès sur des instances

Les prescriptions suivantes s'appliquent pour l'insertion de canaux implicites issus d'un accès sur des ensembles d'instances au sein de l'agent ou du type d'agent vers un accès sur l'agent ou le type d'agent:

- a) si un élément de la liste de signaux de sortie associée à un accès en dehors d'un agent (ou d'un type d'agent) correspond à un élément d'une liste de signaux de sortie associée à un accès d'une instance au sein de l'agent (ou, respectivement, du type d'agent); et
- b) s'il n'existe pas de canal explicite connecté à l'accès en dehors de l'agent (ou, respectivement, du type d'agent) ni de canal explicite connecté à l'accès de l'instance au sein de l'agent (ou, respectivement, du type d'agent)

alors:

- i) s'il n'existe pas de canal implicite entre les deux accès en direction de l'accès situé en dehors de l'agent (ou, respectivement, du type d'agent), un canal implicite unidirectionnel est alors créé depuis l'accès de l'instance au sein de l'agent (ou, respectivement, du type d'agent) vers l'accès situé en dehors de l'agent (ou, respectivement, du type d'agent); ce canal est sans retard s'il se trouve au sein d'un processus (ou d'un type de processus), il est avec retard dans le cas contraire; et
- ii) l'élément est ajouté à la liste de signaux du canal implicite."

4.91 Erreur d'écriture – Paragraphe 10.2: Connexion, Grammaire graphique concrète

Remplacer:

"Sur le plan graphique, la construction connexion est représentée par la connexion graphique d'un symbole <channel symbol> dans une zone <channel definition area> à un identificateur <external channel identifiers> dans un diagramme <gate on diagram>."

par:

"Sur le plan graphique, la construction connexion est représentée par la connexion graphique d'un symbole <channel symbol> dans une zone <channel definition area> à des identificateurs <external channel identifiers> connectés au symbole de trame qui les englobe."

4.92 Erreur d'écriture – Paragraphe 10.2: Connexion, Modèle

Ajouter l'alinéa suivant à la fin du paragraphe:

"Lorsqu'un diagramme est contenu directement dans d'autres diagrammes (non référencés), chaque identificateur <external channel identifiers> est alors omis, compte tenu du fait que les canaux externes connectés au même point de la trame du diagramme à partir de l'extérieur du diagramme sont indiqués directement."

4.93 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 10.3: Signal, Grammaire textuelle concrète

Remplacer la règle <procedure reference> par:

<signal reference> ::=

<type preamble>

signal <type reference heading> <type reference properties>

en la faisant suivre du nouvel alinéa:

"Un en-tête <type reference heading> qui fait partie d'une règle <signal reference> doit posséder un nom <signal name>."

4.94 Clarification – Paragraphe 10.3: Signal, Grammaire graphique concrète

Remplacer le texte "<graphical type reference heading> qui contient" par le texte "<type reference heading> avec" de manière à obtenir l'alinéa révisé suivant:

La zone <type reference area> qui fait partie d'une zone <signal reference area> doit avoir un en-tête <type reference heading> qui contient un nom <signal name>.

4.95 Erreur d'écriture – Interface absente – Paragraphe 10.4: Définition de liste de signaux, Grammaire textuelle concrète

Dans l'alinéa débutant par:

"Un élément <signal list item> qui est un identificateur <identifieur> désigne un identificateur <signal identifieur> ou <timer identifieur> ..."

remplacer "ou <timer identifieur>" par ", <timer identifieur> ou <interface identifieur>".

4.96 Erreur d'écriture – Interface absente – Paragraphe 10.4: Définition de liste de signaux, Grammaire textuelle concrète

Supprimer l'alinéa:

"Il est permis d'utiliser **this** dans les listes <signal list> qui font partie des contraintes <gate constraint> (voir 8.1.6)."

L'alinéa est erroné parce que la syntaxe ne permet pas l'utilisation du mot clé **this**.

4.97 Erreur d'écriture – Exception absente – Paragraphe 10.5: Procédures distantes, Grammaire textuelle concrète

Ajouter un élément "<on exception>" optionnel à la règle "<remote procedure call>" de manière à obtenir la règle:

```
<remote procedure call> ::=  
    call <remote procedure call body> [ <on exception> ]
```

4.98 Erreur d'écriture – Définition de l'ensemble de sortie complet – Paragraphe 10.5: Procédures distantes, Grammaire textuelle concrète

Dans l'alinéa suivant, remplacer "Annexe F" par "le paragraphe 8.1.1.1 et le paragraphe 9" (se référer à la modification concernant le paragraphe 8.1.1.1):

"Une procédure distante mentionnée dans un appel <remote procedure call> doit être dans l'ensemble de sortie complet (voir Annexe F) d'un type d'agent ou d'un ensemble d'agents englobant."

4.99 Erreur d'écriture – Définition de l'ensemble de sortie complet – Paragraphe 10.5: Procédures distantes, Grammaire textuelle concrète

La spécification d'une définition de procédure ne doit pas contenir le mot clé **nodelay**.

Supprimer "[**nodelay**]" dans la syntaxe de la règle <remote procedure definition>.

Supprimer la phrase: "L'utilisation de **nodelay** est décrite en 10.1."

4.100 Erreur d'écriture – Retour supplémentaire, tabulation supplémentaire – Paragraphe 10.5: Procédures distantes, Modèle

Dans le SDL:

```
decision newn = n;  
(true):  
(false): nextstate pWAIT;  
enddecision;  
return;
```

supprimer la ligne:

```
return;
```

Supprimer une tabulation au début de la ligne avec deux tabulations faisant suite au texte "de procédure distante:"

```
input pCALL(fpar,n);
```

4.101 Erreur d'écriture – Nodelay – Paragraphe 10.5: Procédures distantes, Modèle

La spécification d'une définition de procédure distante ne doit pas contenir le mot clé **nodelay**.

Supprimer la phrase:

"Lorsqu'une procédure distante est acheminée sur des canaux explicites, le mot clé **nodelay** de la définition <remote procedure definition> est ignoré."

4.102 Erreur d'écriture – Exception absente – Paragraphe 10.6: Variables distantes, Grammaire textuelle concrète

Remplacer la règle <export> par les deux règles suivantes:

<export> ::=

export <export body> [<on exception>]

<export body> ::=

(<variable identifier> { , <variable identifier> }*)

4.103 Erreur d'écriture – Référence à l'Annexe F et nodelay – Paragraphe 10.6: Variables distantes

Une spécification de variable distante ne doit pas contenir le mot clé **nodelay**.

Supprimer (deux fois) "[**nodelay**]" dans la syntaxe de la règle <remote variable definition>.

Supprimer la phrase:

"L'utilisation de **nodelay** est décrite dans l'Annexe F."

Remplacer dans l'alinéa qui suit "Annexe F" par "8.1.1.1 et le paragraphe 9".

Supprimer la phrase:

"Une variable distante mentionnée dans une expression <import expression> doit être dans l'ensemble de sortie complet (voir Annexe F) d'un type d'agent ou d'un ensemble d'agents englobant."

Dans le paragraphe *Modèle* supprimer la phrase:

"Lorsqu'une variable distante est véhiculée sur des canaux explicites, le mot clé **nodelay** figurant dans la règle <remote variable definition> est ignoré."

4.104 Erreur d'écriture – Erreur de présentation – Paragraphe 11.2: Etat, Grammaire abstraite

Ajouter la règle *Connect-node-set* la règle *State-node*

State-node

:: *State-name*
 [On-exception]
 Save-signalset
 Input-node-set
 Spontaneous-transition-set
 Continuous-signal-set
 Connect-node-set
 [*Composite-state-type-identifier*]

4.105 Clarification – Présentation d'état de base et d'état composite – Paragraphe 11.2: Etat, Grammaire abstraite

Remplacer la phrase:

"Un nœud *State-node* ayant un identificateur *Composite-state-type-identifier* représente une application <composite state application>."

par:

"Un nœud *State-node* ayant un identificateur *Composite-state-type-identifier* représente un état composite <composite state application>."

et insérer les deux alinéas suivants avant la Grammaire graphique concrète (pour l'application de l'état composite):

"La partie <connect part> est autorisée uniquement pour un état <state> avec une liste <state list> qui contient un élément <composite state item>.

Un élément <composite state item> ou un état <typebased composite state> contiendra uniquement des paramètres <actual parameters> s'il s'agit d'une zone <state area> qui coïncide avec une zone <nextstate area>. Dans ce cas, la zone <state area> doit contenir uniquement un nom <composite state name> et, de manière optionnelle, des paramètres <actual parameters>."

4.107 Clarification – Présentation d'état de base et d'état composite – Paragraphe 11.2: Etat, Grammaire graphique concrète

Remplacer le nom de la règle "<basic state area>" par "<state area>" et ajouter une nouvelle variante associée "<connect association area>" à la fin de la règle de manière à obtenir la règle suivante (qui remplace également la règle "<composite state application>"):

<state area> ::=

```
<state symbol> contains <state list>
[ is connected to <on exception association area> ]
is associated with
{
    <input association area>
    | <priority input association area>
    | <continuous signal association area>
    | <spontaneous transition association area>
    | <save association area>
    | <connect association area> }*
```

Ajouter à la fin de la syntaxe, immédiatement avant l'alinéa débutant par "Un état <state are> représente ...", la règle suivante (de l'application d'état composite):

<connect association area> ::=

<solid association symbol> **is connected to** <connect area>

Ajouter l'alinéa suivant avant le paragraphe *Sémantique*:

"La règle <connect association area> est autorisée uniquement pour une zone <state area> avec une liste <state list> contenant un élément <composite state item>."

4.108 Clarification – Présentation d'état de base et d'état composite – Paragraphe 11.2: Etat, Sémantique

Ajouter les deux alinéas suivants au début du paragraphe *Sémantique*:

"Un état représente soit un état de base, soit une application d'état composite.

La sémantique de l'application d'état composite est fournie dans le paragraphe 11.11."

Remplacer au début de l'alinéa suivant "Un état représente ..." par "Un état de base représente ...".

4.109 Clarification – Sélection de transition – Paragraphe 11.2.1: Etat de base, Sémantique

Remplacer:

"... dans l'ordre suivant:

a) ..."

par:

"... avec les étapes suivantes. L'ensemble de signaux considéré est mis à jour avec les signaux de l'accès d'entrée lors de chaque itération des étapes; dans le cas contraire, le même ensemble est pris en compte lors de chaque étape.

a) ..."

et remplacer l'élément d) par:

"d) si aucun signal validé n'a été trouvé, les étapes précédentes sont alors itérées dès que les signaux sur le port d'entrée diffèrent de l'ensemble de signaux déjà pris en compte, ou s'il existe un nœud *Input-node* dont l'expression *Provided-expression* est susceptible d'avoir été modifiée ou dont l'expression *Continuous-expression* est susceptible d'avoir été modifiée. Une expression *Provided-expression* ou *Continuous-expression* peut changer de valeur seulement si elle contient une expression *NOW-expression*, *Timer-active-expression*, *Any-expression* ou un accès *Variable-access* vers une variable définie dans un processus englobant qui est modifiée par une assignation dans une autre instance d'agent ou une autre partition d'état."

4.110 Clarification – Présentation d'état de base et d'état composite – Paragraphe 11.2.1: Etat de base, Modèle

Dans le premier paragraphe du *Modèle*, remplacer "<state name>, une copie ..." par "un élément <state name>, une copie ...".

Dans le troisième paragraphe du *Modèle*, ajouter après "un pour chaque nom <state name>" le texte "et nom <composite state name>" et ajouter après "les noms <state name>" le texte "et les noms <composite state name>".

Supprimer la totalité de l'ancien paragraphe 11.2.2, **Application d'état composite**.

4.111 Clarification – Remplacer <action statement> par <action> – Paragraphe 11.3: Entrée, Modèle

Remplacer "<action statement>" par "<action>" (se référer à la modification 4.155).

4.112 Erreur d'écriture – On-exception omis – Paragraphe 11.5: Signal continu, Grammaire abstraite

Remplacer:

Continuous-signal :: *Continuous-expression* [*Priority-name*] *Transition*

par:

Continuous-signal :: [*On-exception*]
Continuous-expression
[*Priority-name*]
Transition

4.113 Clarification – Sélection de transition – Paragraphe 11.5: Signal continu, Sémantique

Modifier comme suit le paragraphe *Sémantique*:

"L'expression *Continuous-expression* est interprétée comme faisant partie de l'état auquel est associé le signal *Continuous-signal* (se référer au paragraphe 11.2). Le signal continu est validé si l'expression *Continuous-expression* renvoie la valeur booléenne "Vrai" prédéfinie.

Le signal continu avec la valeur la plus faible pour le nom *Priority-name* possède la priorité la plus élevée."

4.114 Clarification – Sélection de transition – Paragraphe 11.6: Condition de validation, Sémantique

Remplacer le paragraphe *Sémantique* par le texte suivant:

"L'expression *Provided-expression* d'un nœud *Input-node* est interprétée comme faisant partie de l'état auquel est attaché ce nœud (se référer au paragraphe 11.2).

Un signal est validé au niveau de l'accès d'entrée si toutes les expressions *Provided-expressions* d'un nœud *Input-node* renvoient la valeur booléenne "Vrai" prédéfinie, ou si le nœud *Input-node* ne possède pas d'expression *Provided-expression*. L'expression *Provided-expression* d'une transition *Spontaneous-transition* peut être interprétée à tout instant pendant que l'agent se trouve dans cet état."

4.115 Clarification – Paragraphe 11.7: Sauvegarde, Grammaire abstraite

Dans la grammaire:

Save-signalset :: *Signal-identifier-set*

Remplacer "::" par "=" de manière à obtenir

Save-signalset = *Signal-identifier-set*

4.116 Clarification – Zone de corps à la place de zone de graphe – Paragraphe 11.10: Etiquette, Grammaire textuelle concrète

Remplacer l'alinéa après la syntaxe débutant par "Le terme ..." par:

"Le terme "body" (corps) est utilisé pour se rapporter à un graphe de machine à états, éventuellement après transformation à partir d'une liste <statement list> et après transformation à partir d'un type. Un corps se rapporte donc à un corps <agent body>, <procedure body>, <operation body>, une zone <agent body area>, <procedure body area>, <operation body area> ou à un corps <composite state body>.

4.117 Clarification – Remplacer <action statement> par <action> – Paragraphe 11.10: Etiquette, Grammaire textuelle concrète

Remplacer "<action statement>" par "<action>" dans le dernier alinéa avant la grammaire graphique concrète (se référer à la modification 4.155).

4.118 Clarification – Remplacer "graph area" par "body area" – Paragraphe 11.10: Etiquette, Grammaire graphique concrète

Remplacer l'alinéa après la syntaxe débutant par "Une zone <in connector area> ..." par:

par:

"Une zone <in connector area> représente la continuation d'un symbole <flow line symbol> à partir d'une zone <out connector area> correspondante avec le même nom <connector name>, dans la même zone <state machine graph area>, la même zone <agent body area> ou la même zone <procedure body area>."

4.119 Clarification – Remplacer <action statement> par <action> – Paragraphe 11.10: Etiquette, Modèle

Remplacer "<action statement>" par "<action>" (se référer à la modification 4.155).

**4.120 Clarification – Remplacer <terminator statement> par <terminator> –
Paragraphe 11.10: Etiquette, Modèle**

Remplacer "<terminator statement>" par "<terminator>" (se référer à la modification 4.155).

4.121 Erreur d'écriture – Mot "by" absent – Paragraphe 11.11: Machine à états et état composite

[Cetle erreur n'affecte pas la version française]

4.122 Clarification – Paragraphe 11.11: Machine à états et état composite, Sémantique

Dans le dernier alinéa, remplacer l'avant-dernière phrase:

"La procédure de sortie est invoquée après l'interprétation d'un nœud *Return-node* du graphe *Composite-state-graph*, ou lorsqu'une transition associée directement au nœud *State-node* est interprétée."

par:

"La procédure de sortie est invoquée après l'interprétation d'un nœud *Return-node* du graphe *Composite-state-graph* ou avant l'interprétation d'une transition associée directement au nœud *State-node*, si de telles transitions existent."

4.123 Erreur d'écriture – Modèle absent – Paragraphe 11.11: Machine à états et état composite, Modèle

Ajouter le paragraphe *Modèle* suivant:

"Modèle

Un état *Composite-state* possède un type d'état composite anonyme implicite qui définit les propriétés de l'état composite.

Un état *Composite-state* qui est une spécialisation constitue, pour ce type, une notation abrégée qui définit un type d'état composite implicite et un état composite basé sur le type."

4.124 Erreur d'écriture – Nœud de départ optionnel – Paragraphe 11.11.1: Graphe d'état composite, Grammaire abstraite

Ajouter la syntaxe abstraite pour le graphe *State-transition-graph* à partir du paragraphe 9.

Dans la définition de *State-transition-graph*, remplacer "*State-start-node*" par "[*State-start-node*]". La nouvelle syntaxe est la suivante:

```
State-transition-graph      ::      [On-exception]  
                               [State-start-node]  
                               State-node-set  
                               Free-action-set  
                               Exception-handler-node-set
```

Ajouter l'alinéa suivant à la fin de la *Grammaire abstraite*:

"Dans une spécification *SDL-specification*, tous les agents potentiellement instanciables doivent posséder un nœud *State-start-node*."

4.125 Clarification – Paragraphe 11.11.1: Graphe d'état composite, Grammaire textuelle concrète

Remplacer la règle <composite state graph> par les deux règles suivantes:

```
<composite state graph> ::=  
    {<package use clause>}*  
    <composite state heading> <end> <composite state structure>  
    endsubstructure [ [<qualifier>] <composite state name> ] <end>  
  
<composite state structure> ::=  
    substructure  
        [<valid input signal set>]  
        {<gate in definition>}*  
        <state connection points>*  
        <entity in composite state>*  
        { <composite state body> | <state aggregation body> }
```

Ajouter l'alinéa suivant après la nouvelle règle <composite state structure>:

"Une règle <composite state structure> contiendra un corps <state aggregation body> seulement si elle est contenue directement dans un agrégat <state aggregation> ou dans une définition <composite state type definition> avec un en-tête <state aggregation type heading>; elle contient un corps <composite state body> dans le cas contraire. Une règle <composite state structure> qui contient un corps <state aggregation body> n'aura pas d'ensemble <valid input signal set>."

4.126 Erreur d'écriture – Spécialisation omise – Paragraphe 11.11.1: Graphe d'état composite, Grammaire textuelle concrète

Ajouter un élément <specialization> optionnel à la règle:

```
<composite state heading> ::=  
    state [<qualifier>] <composite state name>  
    [<agent formal parameters>] [<specialization>]
```

4.127 Erreur d'écriture – Éléments omis – Paragraphe 11.11.1: Graphe d'état composite, Grammaire textuelle concrète

Ajouter une référence <procedure reference> en variante de la production pour la règle <entity in composite state> après la définition <procedure definition>.

Ajouter un état <composite state> en variante de la production pour la règle <entity in composite state> après la définition <exception definition>.

4.128 Erreur d'écriture – Règle mal placée – Paragraphe 11.11.1: Graphe d'état composite, Grammaire textuelle concrète

Supprimer la règle <composite state reference>, qui est placée de manière plus adéquate dans le paragraphe 11.11.2 Agrégat d'état.

4.129 Clarification – Phrase mal placée – Paragraphe 11.11.1: Graphe d'état composite, Grammaire textuelle concrète

Après l'alinéa:

"Un seul de ces départs <start> doit être non étiqueté. Chaque point d'entrée et de sortie étiqueté supplémentaire doit être défini par un point <state connection points> correspondant."

ajouter l'alinéa suivant reformulé à partir du paragraphe 11.11.3:

"Un élément <start> avec un nom <state entry point name> (départ étiqueté) qui fait partie d'un corps <composite state body> fera référence à un seul des points <state entry point> du graphe

<composite state graph> qui contient directement le corps <composite state body>. Un élément <return> avec un point <state exit point> (retour étiqueté) qui fait partie d'un corps <composite state body> fera référence à un seul des points <state exit point> du graphe <composite state graph> qui contient directement le corps <composite state body>."

4.130 Clarification – Utilisation d'accès – Paragraphe 11.11.1: Graphe d'état composite, Grammaire textuelle concrète

Ajouter la phrase suivante immédiatement avant le paragraphe *Grammaire graphique concrète*:

"Un accès <gate> d'un graphe <composite state graph> peut être un point d'extrémité <channel endpoint> seulement si le graphe <composite state graph> est la division <state partitioning> qui représente la machine d'état d'un agent."

4.131 Clarification – Paragraphe 11.11.1: Graphe d'état composite, Grammaire graphique concrète

Remplacer la règle <composite state graph area> par les deux règles suivantes:

```
<composite state graph area> ::=
    <frame symbol> contains
        { <composite state heading> <composite state structure area> }
    is associated with {<graphical state connection point>* } set
    is connected to { {<gate on diagram> | <external channel identifiers>}* } set
    [ is associated with <package use area> ]
```

```
<composite state structure area> ::=
    {
        <composite state text area>*
        <entity in composite state area>*
        { <composite state body area> | <state aggregation body area> } } set
```

Ajouter la phrase suivante après la nouvelle règle <composite state structure area>:

"Une règle <composite state structure area> contiendra une zone <state aggregation body area> seulement si elle est contenue directement dans une zone <state aggregation area> ou dans un diagramme <composite state type diagram> avec un en-tête <state aggregation type heading>; elle contient une zone <composite state body area> dans le cas contraire."

4.132 Erreur d'écriture – Eléments omis – Paragraphe 11.11.1: Graphe d'état composite, Grammaire graphique concrète

Ajouter la définition "<exception definition>" comme variante dans le symbole <text symbol> pour la production <composite state text area> après la référence "<procedure reference>".

4.133 Clarification – Paragraphe 11.11.1: Graphe d'état composite, Grammaire graphique concrète

Remplacer le nom de la règle "<type in composite state area>" par "<entity in composite state area>", qui est un nom plus correct.

Remplacer l'alinéa:

"Il doit y avoir au plus un ensemble <valid input signal set> dans les zones <composite state text area> d'une zone <composite state graph area>."

par:

"Il doit y avoir au plus un ensemble <valid input signal set> dans les zones <composite state text area> d'une zone <composite state graph area> (ou dans la définition correspondante de type d'état composite). Un ensemble <valid input signal set> ne doit pas être contenu dans une zone <composite

state text area> d'une zone <state aggregation area> (ou dans la définition correspondante de type d'état composite)."

4.134 Clarification – Phrase mal placée – Paragraphe 11.11.1: Graphe d'état composite, Grammaire graphique concrète

Ajouter les alinéas suivants (après l'alinéa ci-dessus):

"Une seule des zones <start area> au plus doit être dépourvue d'étiquette. Chaque point d'entrée ou de sortie supplémentaire dépourvu d'étiquette doit être défini par un point <state connection points> correspondant. Chaque zone <start area> supplémentaire contiendra un nom <state entry point name> différent.

Une zone <start area> avec un nom <state entry point name> (départ étiqueté) faisant partie d'une zone <composite state body area> fera référence à l'un seul des points <state entry point> de la zone <composite state graph area> qui contient directement la zone <composite state body area>. Une zone <return area> avec un point <state exit point> (sortie étiquetée) faisant partie d'une zone <composite state body area> fera référence à un seul des points <state exit point> de la zone <composite state graph area> qui contient directement la zone <composite state body area>."

4.135 Clarification – Paragraphe 11.11.1: Graphe d'état composite, Sémantique

Modifier comme suit le dernier alinéa du paragraphe *Sémantique*:

"Il est possible de spécifier un état <composite state> qui consiste uniquement en transitions reçues dans l'état astérisque, sans élément <start> et sans sous-état. Ces transitions peuvent se terminer par un état <dash nextstate> ou un élément <return>. Elles s'appliquent lorsque l'agent ou la procédure se trouve dans l'état composite. L'état suivant d'une telle transition qui se termine par l'état <dash nextstate> est l'état composite. Toutefois, les procédures des définitions Exit-procedure-definition et Entry-procedure-definition de l'état composite ne sont pas appelées."

4.136 Clarification – Paragraphe 11.11.2: Agrégat d'état

Dans la deuxième phrase, remplacer "est entrelacée au niveau des transitions" par "de transitions alternées" et supprimer la dernière phrase: "Les agrégats d'état peuvent être utilisés pour diviser le graphe d'un état."

4.137 Clarification – Conditions – Paragraphe 11.11.2: Agrégat d'état, Grammaire abstraite

La combinaison de conditions dans la *Grammaire abstraite* et dans la *Grammaire graphique concrète* n'est pas claire. Les conditions doivent être déplacées de manière à figurer uniquement dans la *Grammaire abstraite*.

Dans la *Grammaire abstraite*, remplacer:

"Tous les points d'entrée et de sortie de l'état conteneur et des divisions d'état doivent apparaître dans une seule et unique définition *Connection-definition*."

par:

"Pour chaque division *State-partition*, chacun des points d'entrée de l'état conteneur figurera dans une et une seule définition *Connection-definition*. Pour chaque division *State-partition*, chacun des points de sortie du conteneur figurera dans une et une seule définition *Connection-definition*."

4.138 Clarification – Paragraphe 11.11.2: Agrégat d'état, Grammaire abstraite

Remplacer "*State-partition-set*" par "*State-partitions*" dans la phrase:

"Les ensembles de signaux d'entrée dans des divisions *State-partition-set* au sein d'un état composite doivent être disjoints."

de manière à obtenir la phrase:

"Les ensembles de signaux d'entrée dans des divisions *State-partitions* au sein d'un état composite doivent être disjoints."

4.139 Clarification – Paragraphe 11.11.2: Agrégat d'état, Grammaire textuelle concrète

Modifier comme suit la règle <state aggregation>:

```
<state aggregation> ::=
    {<package use clause>}*
    <state aggregation heading> <end> <composite state structure>
    endsubstructure [ [<qualifier>] <composite state name> ] <end>
```

4.140 Erreur d'écriture – Spécialisation omise – Paragraphe 11.11.2: Agrégat d'état, Grammaire textuelle concrète

Ajouter un élément <specialization> optionnel à la règle:

```
<state aggregation heading> ::=
    state aggregation [<qualifier>] <composite state name>
    [<agent formal parameters>][<specialization>]
```

4.141 Clarification – Paragraphe 11.11.2: Agrégat d'état, Grammaire textuelle concrète

Dans la règle <state aggregation body>, remplacer "<state partitioning>" par "<state partition>".

Remplacer le nom de la règle "<state partitioning>" par "<state partition>".

4.142 Erreur d'écriture – Autoriser un corps vide – Paragraphe 11.11.2: Agrégat d'état, Grammaire textuelle concrète

Dans la règle <state aggregation body>, remplacer "+" par "*" de manière à autoriser un nombre de répétitions supérieur ou égal à zéro comme suit:

```
<state aggregation body> ::=
    {
        <state partition>
        | <state partition connection> }*
```

4.143 Clarification – Référence d'état composite – Paragraphe 11.11.2: Agrégat d'état, Grammaire textuelle concrète

La règle <composite state reference> est déplacée du paragraphe 11.11.1 vers cet emplacement, avec la syntaxe suivante:

```
<composite state reference> ::=
    state substructure <composite state name> referenced <end>
```

4.144 Erreur d'écriture – Référence d'état composite – Paragraphe 11.11.2: Agrégat d'état, Grammaire textuelle concrète

Les identificateurs des points d'entrée doivent être des états composites et non des divisions d'état. La syntaxe révisée est la suivante:

```
<outer entry point> ::=
    <composite state identifier> via <point>
```

<inner entry point> ::=
 <composite state identifier> **via** <point>

4.145 Clarification – Paragraphe 11.11.2: Agrégat d'état, Grammaire graphique concrète

Modifier comme suit la règle <state aggregation area>:

<state aggregation area> ::=
 <frame symbol> **contains** {
 <state aggregation heading>
 <composite state structure area>
 is associated with {<graphical state connection point>* } **set**
 is connected to { {<gate on diagram> | <external channel identifiers>}* } **set**
 [**is associated with** <package use area>]

4.146 Erreur d'écriture – Autoriser un corps vide – Paragraphe 11.11.2: Agrégat d'état, Grammaire graphique concrète

Dans la règle <state aggregation body area>, remplacer "<connection definition area>" par "<state partition connection area>" et remplacer "+" par "*" de manière à autoriser un nombre de répétitions supérieur ou égal à zéro:

<state aggregation body area> ::=
 { { <state partition area> | <state partition connection area>* } } **set**

(La règle <connection definition area> est renommée <state partition connection area> – voir ci-dessous.)

4.147 Clarification – Paragraphe 11.11.2: Agrégat d'état, Grammaire graphique concrète

Dans la règle <state partition area> remplacer "<state partition reference area>" par "<composite state reference area>".

Remplacer le nom de la règle "<state partition reference area>" par "<composite state reference area>".

4.148 Erreur d'écriture – Identificateur – Paragraphe 11.11.2: Agrégat d'état, Grammaire graphique concrète

Modifier la règle <inherited state partition definition> de manière à faire référence à un identificateur <composite state identifier> et non à un identificateur <state partition identifier>.

4.149 Erreur d'écriture – Syntaxe de points de connexion – Paragraphe 11.11.2: Agrégat d'état, Grammaire graphique concrète

Remplacer les règles:

<connection definition area> ::=
 <solid association symbol> **is connected to** <graphical point>

<graphical point> ::=
 <graphical state connection point>
 | { <state entry points> | <state exit points> } **is associated with** <state partition
area> | <state connection point symbol> **is connected to** <frame symbol>

par les règles suivantes, dans lesquelles la règle <connection definition area> est renommée "<state partition connection area>":

<state partition connection area> ::=
 <solid association symbol> **is connected to**
 [<outer graphical point> <inner graphical point>]

<outer graphical point> ::=
 { <state entry points> | <state exit points> } **is associated with** <frame symbol>
 <inner graphical point> ::=
 { <state entry points> | <state exit points> } **is associated with** <state partition area>

4.150 Erreur d'écriture – Accès omis – Paragraphe 11.11.2: Agrégat d'état, Grammaire graphique concrète

Il a été oublié d'autoriser un accès dans la syntaxe graphique des états (bien que de tels accès aient été autorisés dans la forme textuelle correspondante). Les règles <composite state reference area> et <graphical type based state partition definition> sont mises à jour comme suit et le texte ci-dessous est ajouté à la suite des règles:

<composite state reference area> ::=
 <state symbol> **contains** { <state name> {<gate>*}**set** }
 <graphical typebased state partition definition> ::=
 <state symbol> **contains** { <typebased state partition heading> {<gate>*}**set** }

Le ou les accès <gate> contenus dans un ou plusieurs symboles <state symbol> sont placés au voisinage de la frontière des symboles et sont associés au point de connexion vers des canaux.

Un accès <gate> est autorisé dans un symbole <state symbol> d'une définition <graphical typebased state partition definition> ou d'une zone <composite state reference area> d'une zone <state partition area> seulement si la zone <state partition area> représente la machine d'état d'un agent ou d'un type d'agent.

4.151 Clarification – Conditions – Paragraphe 11.11.2: Agrégat d'état, Grammaire graphique concrète

La combinaison de conditions dans la *Grammaire abstraite* et dans la *Grammaire graphique concrète* n'est pas claire. Les conditions doivent être déplacées de manière à figurer uniquement dans la *Grammaire abstraite*.

Supprimer les conditions suivantes dans la *Grammaire graphique concrète*:

"Le même point de sortie d'état d'une division d'état ne doit pas être relié à plus d'un point de sortie d'état de l'état englobant.

Un point d'entrée d'état de l'état englobant ne doit pas être relié à deux points d'entrée d'état différents de la même division d'état."

4.152 Clarifications – Paragraphe 11.11.2: Agrégat d'état, Sémantique

Dans la quatrième phrase du premier alinéa, remplacer "la création des divisions *State-partition-set* contenues et leurs connexions" par "la création de chaque division *State-partition* contenue et de sa connexion".

4.153 Clarification – Phrase mal placée – Paragraphe 11.11.3: Point de connexion d'état, Sémantique

Supprimer la phrase suivante dans le paragraphe 11.11.3 (les ajouts au paragraphe 11.11.1 et le texte du paragraphe 11.12.2.4 ont le même effet):

"Un état <composite state> ne peut correspondre qu'à ses propres points <state entry point> dans des départs <start> étiquetés et à ses propres points <state exit point> dans des retours <return> étiquetés."

4.154 Extension – Liste de connexions plus flexible – Paragraphe 11.11.4: Connexion

Les règles <connect list> et <asterisk connect list> ont été modifiées pour rendre la syntaxe plus flexible et la règle <state exit point list> a été ajoutée comme suit:

```
<connect list> ::=
    <state exit point list>
    | <asterisk connect list>

<state exit point list> ::=
    { <state exit point name> | default } { , { <state exit point name> | default } } *

<asterisk connect list> ::=
    <asterisk> [ ( <state exit point list> ) ]
```

Le texte du paragraphe *Modèle* est remplacé par:

"Le mot clé **default** dans une règle <state exit point list> représente un élément <return> non étiqueté.

Lorsque la liste <connect list> d'une partie <connect part> contient plus d'un nom <state exit point name>, une copie de la partie <connect part> est alors créée pour chacun de ces noms <state exit point name>. La partie <connect part> est remplacée ensuite par ces copies.

Une liste <connect list> qui contient une liste <asterisk connect list> est transformée en une liste de points <state exit point>, à raison d'une telle liste pour chaque point <state exit point> de l'état <composite state> concerné (y compris pour l'élément <return> non étiqueté), à l'exception de ceux qui sont mentionnés entre parenthèses après l'élément <asterisk>. La liste de points <state exit point> est transformée ensuite comme décrit précédemment."

4.155 Erreur d'écriture – Traitement d'exception – Paragraphe 11.12.1: Corps de transition, Grammaire textuelle concrète

Dans la *Grammaire textuelle concrète*, changer le nom des instructions "<terminator statement>" et "<action statement>" respectivement en "<terminator>" et "<action>"; éliminer les règles <action 1>, <action 2>, <terminator 1> et <terminator 2> de manière à obtenir les règles syntaxiques suivantes:

```
<transition> ::=
    { <transition string> [ <terminator> ] }
    | <terminator>

<transition string> ::=
    { <action> } +

<action> ::=
    [ <label> ]
    {
        <task>
        <output>
        <create request>
        <decision>
        <set>
        <reset>
        <export>
        <procedure call>
        <remote procedure call>
        <transition option> } <end>

<terminator > ::=
    [ <label> ]
    {
        <return>
        <raise>
        <nextstate>
        <join>
        <stop> } end>
```

et remplacer "<terminator statement>" par "<terminator>" dans le texte qui suit les règles syntaxiques.

Modifier comme suit les règles dans la *Grammaire graphique concrète*:

<transition area> ::=
[<transition string area> **is followed by**]
<terminator area>

<terminator area> ::=
| <state area>
| <nextstate area>
| <decision area>
| <stop symbol>
| <merge area>
| <out connector area>
| <return area>
| <transition option area>
| <raise area>

<transition string area> ::=
<action area>
[**is followed by** <transition string area>]

<action area> ::=
| <task area>
| <output area>
| <create request area>
| <procedure call area>
| <remote procedure call area>

4.156 Erreur d'écriture – Corps d'opérateurs – Paragraphe 11.12.1: Corps de transition, Grammaire graphique concrète

Ajouter l'alinéa suivant à la fin du paragraphe:

"Une zone <transition area> faisant partie d'une zone <operation body area> ne contiendra pas de zone <state area> ou <nextstate area>."

4.157 Erreur d'écriture – Référence à l'Annexe F – Paragraphe 11.12.1: Corps de transition, Sémantique

Dans l'alinéa débutant par:

"Une transition dans un processus ou dans un bloc peut ..."

remplacer "par les règles dans l'Annexe F" par "par les règles données dans les paragraphes *Modèle* de la présente Recommandation."

4.158 Clarification – Remplacer <action statement> par <action> – Paragraphe 11.12.2.2: Branchement

Dans la première phrase, remplacer "<action statement>" par "<action>" (se référer à la modification 4.155).

4.159 Clarification – <agent body> – Paragraphe 11.12.2.2: Branchement, Grammaire textuelle concrète

Pour rester en accord avec la clarification de suppression du type dans les corps, remplacer le texte "<agent type body>" par "<agent body> dans une définition de type" dans la phrase qui fait suite à la syntaxe.

4.160 Clarification – <agent body area> – Paragraphe 11.12.2.2: Branchement, Grammaire graphique concrète

Pour rester en accord avec la représentation PR et d'autres clarifications, remplacer (deux fois) "<state machine graph area>" par "<agent body area>" dans la première phrase qui suit la syntaxe.

4.161 Extension – Autoriser un arrêt dans les procédures, etc. – Paragraphe 11.12.2.3: Arrêt

Se référer également aux paragraphes 11.14 et 11.14.2 Autoriser un arrêt dans les procédures etc.

Dans la *Grammaire abstraite*, supprimer la condition "Un nœud *Stop-node* ne doit pas être situé à l'intérieur d'un graphe *Procedure-graph*."

Ajouter à la fin du paragraphe *Sémantique*

"L'interprétation d'un nœud *Stop-node* dans un graphe *Procedure-graph* ou *State-transition-graph* conduit à l'arrêt de l'agent qui interprète ce graphe *Procedure-graph*. L'interprétation de la procédure, de l'opération, de l'instruction composite ou de l'état composite prend fin et la condition d'arrêt se propage vers l'extérieur à destination de l'appelant; cet arrêt est traité comme si un nœud *Stop-node* avait été interprété à la place de l'appel de procédure, de l'application de l'opération, de l'invocation de l'instruction composite ou de l'entrée dans l'état composite. La terminaison se propage vers l'extérieur jusqu'à ce qu'elle parvienne à l'agent conteneur."

4.162 Erreur d'écriture – Sorte de retour – Paragraphe 11.12.2.4: Retour, Grammaire abstraite

Ajouter à la fin de la *Grammaire abstraite*:

"La sorte de *Expression* dans un nœud *Value-return-node* doit être compatible avec la sorte du résultat *Result* de la *Procedure* qui l'englobe."

4.163 Erreur d'écritures – Cohérence entre les représentations PR et GR – Paragraphe 11.12.2.4: Retour, Grammaires concrètes

Modifier comme suit la règle <return> dans la *Grammaire textuelle concrète*:

```
<return> ::=
                return [<return body>] [ <end> <on exception> ]
```

Modifier comme suit la règle <return area> dans la *Grammaire graphique concrète*:

```
<return area> ::=
                <return symbol>
                [ is connected to <on exception association area> ]
                [ is associated with <return body> ]
```

et ajouter:

```
<return body> ::=
                <expression>
                | via <state exit point>
```

4.164 Erreur d'écriture – Sorte de retour – Paragraphe 11.12.2.4: Retour, Grammaire graphique concrète

Ajouter à la fin de la *Grammaire graphique concrète*:

"L'élément <expression> dans une règle <return> ou <return area> ne sera pas omis si le domaine de validité qui l'englobe est un opérateur ou une méthode qui fournit un résultat <operation result> ou une procédure qui renvoie une valeur avec un résultat <procedure result> sans nom <variable name>."

NOTE – Si l'élément <expression> est omis dans un opérateur ou une méthode qui fournit un résultat <operation result> ou dans une procédure qui renvoie une valeur avec un résultat <procedure result> nommé, le modèle du paragraphe 9.4 ajoute alors la variable de résultat de la procédure comme élément <expression>."

4.165 Erreur d'écriture – Sorte de retour – Paragraphe 11.12.2.4: Retour, Sémantique

Remplacer l'alinéa:

"c) si un nœud *Value-return-node* est interprété, le résultat de l'*Expression* est renvoyé dans le contexte appelant;"

par:

"c) si un nœud *Value-return-node* est interprété, le résultat de l'entité *Expression* est alors interprété de la même manière que pour une entité *Expression* assignée à une variable avec la sorte du résultat (se référer au paragraphe 12.3.3), mais sans que le résultat soit associé à une variable et sans vérification du domaine de valeur; l'objet ou la valeur du résultat est renvoyé ensuite au contexte appelant."

4.166 Erreur d'écriture – Traitement d'exception – Paragraphe 11.13.1: Tâche, Grammaire textuelle concrète

Modifier comme suit la règle <task>:

```
<task> ::=
        task <textual task body> [ <end> <on exception> ]
```

4.167 Clarification et modification – Paragraphe 11.13.1: Tâche

Dans la *Grammaire textuelle concrète*, modifier comme suit la dernière variante dans la règle <textual task body>

```
| [ <comment body> ] <left curly bracket> <statement list> <right curly bracket>
```

Dans la *Grammaire graphique concrète*, modifier comme suit la règle <task area>:

```
<task area> ::=
        <task symbol> contains <graphical task body>
        [ is connected to <on exception association area> ]
        | <macro symbol> contains { <macro name> [<macro call body>] }
```

et ajouter à la fin des règles:

"<macro symbol> ::=



A la fin de la *Grammaire graphique concrète*, insérer le texte suivant avant le paragraphe *Sémantique*:

"Une règle <task area> ou <task> qui contient un élément <assignment> représente une entité *Assignment* ou *Assignment-attempt*.

Une règle <task area> ou <task> qui contient un seul élément <set> représente un nœud *Set-node*.

Une règle <task area> ou <task> qui contient un seul élément <reset> représente un nœud *Reset-node*.

Une règle <task area> ou <task> qui contient toute autre liste <statement list> représente un nœud *Compound-node*. Le nom *Connector-name* est représenté par un nom anonyme nouvellement créé. L'ensemble *Variable-definition-set* est représenté par la liste de toutes les définitions <variable definition> dans la liste <statement list>. L'entité *Transition* est représentée par la transformation des

instructions <statements> en liste <statement list> ou par la transformation des instructions <statements> en liste <statement list> suivie d'un nœud *Break-node* avec un nom *Connector-name*, si la liste <statement list> est sans terminaison."

Insérer après le premier alinéa du paragraphe *Sémantique* le nouvel alinéa suivant:

"L'interprétation d'un nœud *Compound-node* est décrite dans le paragraphe 11.14.1. L'interprétation d'une instruction *Assignment* ou *Assignment-attempt* est décrite dans le paragraphe 12.3.3. L'interprétation d'un nœud *Set-node* ou *Reset-node* est décrite dans le paragraphe 11.15."

Remplacer la totalité du paragraphe *Modèle*, y compris la Note, par le texte suivant:

"Si la liste <statement list> d'un corps <textual task body> ou <graphical task body> est vide, la règle <task> ou <task area> correspondante est alors supprimée. Tout élément syntactique figurant en tête d'une telle règle <task> ou <task area> vide conduira dans ce cas directement à l'élément qui fait respectivement suite à la règle <task> ou <task area>."

Un symbole <macro symbol> contenant <macro name> [<macro call body>] est transformé en un symbole <task symbol> contenant **macro** <macro name> [<macro call body>] <end>."

4.168 Erreur d'écriture – Traitement d'exception – Paragraphe 11.13.2: Création, Grammaire textuelle concrète

Modifier comme suit la règle <create request>:

<create request> ::=
create <create body> [<end> <on exception>]

4.169 Clarification – Détermination statique d'ensembles d'instances – Paragraphe 11.13.2: Création, Modèle

L'alinéa c) du *Modèle* impliquait que des ensembles d'instances peuvent être créés de manière dynamique lorsqu'il n'est pas possible de déduire les canaux implicites vers l'ensemble d'instances de la même manière que pour des ensembles d'instances statiques. La possibilité de création d'une instance est, par contre, déterminée de manière statique, ce qui fait qu'un ensemble d'instances (sans instances initiales) est utilisé de manière implicite si aucun ensemble d'instances n'est donné de manière explicite. Le texte de l'alinéa c) du modèle doit être modifié comme suit:

- "c) S'il n'existe pas d'ensemble d'instances du type d'agent indiqué dans l'agent conteneur, alors:
- un ensemble d'instances du type donné avec un nom non ambigu est utilisé de manière implicite dans l'agent conteneur; et
 - l'identificateur <agent type identifier> dans la demande <create request> constitue la syntaxe dérivée pour cet ensemble d'instances implicite."

4.170 Erreur d'écriture – Incohérence opération/procédure – Paragraphe 11.13.3: Appel de procédure, Grammaire abstraite

Remplacer l'alinéa:

"Chaque *Expression* correspondant par sa position à un paramètre *Inout-parameter* ou *Out-parameter* doit être un identificateur *Variable-identifier* dont la sorte est compatible avec la sorte identifiée par l'identificateur *Sort-reference-identifier* du paramètre *Procedure-formal-parameter*."

par:

"Chaque *Expression* correspondant par sa position à un paramètre *Inout-parameter* ou *Out-parameter* doit être un identificateur *Variable-identifier* avec le même identificateur *Sort-reference-identifier* que le paramètre *Procedure-formal-parameter*."

4.171 Erreur d'écriture – Traitement d'exception – Paragraphe 11.13.3: Appel de procédure, Grammaire textuelle concrète

Modifier comme suit la règle <procedure call>:

<procedure call> ::=

call <procedure call body> [<end> <on exception>]

4.172 Erreur d'écriture – Incohérence opération/procédure – Paragraphe 11.13.3: Appel de procédure, Sémantique

Le paragraphe *Sémantique* est modifié comme suit:

"L'interprétation d'une procédure *Call-node* ou *Value-returning-call-node* interprète les expressions des paramètres réels dans l'ordre donné. Si aucune exception n'est activée par l'interprétation des paramètres, l'interprétation est transférée ensuite vers la définition de procédure référencée par l'identificateur *Procedure-identifier* et le graphe de procédure est interprété (l'explication est donnée en 9.4).

Si une <expression> est omise dans les paramètres <actual parameters>, le paramètre formel correspondant ne possède pas d'élément de données associé; c'est-à-dire qu'il est "non défini".

Si la sorte d'argument du nœud *Call-node* ou *Value-returning-call-node* pour un paramètre *In-parameter* ou *Inout-parameter* de la procédure est un type syntype, le contrôle du domaine de valeurs défini dans le paragraphe 12.1.9.5 est alors appliqué au résultat de *Expression*. Si le contrôle du domaine de valeurs fournit la valeur booléenne "Faux" au moment de l'interprétation, l'exception prédéfinie *OutOfRange* (se référer au paragraphe D.3.16) est alors activée et vient remplacer la suite de l'interprétation des paramètres effectifs ou de la définition de procédure.

Si l'exception *OutOfRange* n'est pas activée, l'interprétation de la transition contenant un nœud *Call-node* se poursuit alors lorsque l'interprétation de la procédure appelée est terminée.

Si l'exception *OutOfRange* n'est pas activée, l'interprétation de la transition contenant un nœud *Value-returning-call-node* se poursuit alors lorsque l'interprétation de la procédure appelée est terminée. Le résultat de la procédure appelée est renvoyé par le nœud *Value-returning-call-node*.

Un nœud *Value-returning-call-node* possède une sorte, qui est celle du résultat fourni par l'interprétation de la procédure.

Si la sorte du résultat d'un appel de procédure renvoyant un résultat est un type syntype, le contrôle du domaine de valeurs défini dans le paragraphe 12.1.9.5 est appliqué au résultat de l'appel de la procédure. Si le contrôle du domaine de valeurs fournit la valeur booléenne "Faux" au moment de l'interprétation, l'exception prédéfinie *OutOfRange* est activée (se référer au paragraphe D.3.16)."

4.173 Erreur d'écriture – Traitement d'exception – Paragraphe 11.13.4: Sortie, Grammaire textuelle concrète

Modifier comme suit la règle <output>:

<output> ::=

output <output body> [<end> <on exception>]

4.174 Extension – Types de donnée en ligne – Paragraphe 11.13.4: Sortie, Grammaire textuelle concrète

Modifier comme suit la règle <destination>:

<destination> ::=

<pid expression0> | <agent identifier> | **this**

Remplacer chaque expression "<pid expression>" correspondante par l'expression "<pid expression0>" dans le texte de la *Grammaire textuelle concrète* (5 fois) et dans le paragraphe *Sémantique* (une fois).

4.175 Correction d'erreur – Evaluation de réponses de décision – Paragraphe 11.13.5: Décision

Remplacer la phrase suivante dans la *Grammaire abstraite*:

"Les conditions *Range-condition* des réponses *Decision-answer* doivent s'exclure mutuellement et les expressions *Constant-expression* des conditions *Range-condition* doivent avoir des sortes compatibles."

par:

"Les expressions *Constant-expression* des conditions *Range-condition* doivent avoir des sortes compatibles."

4.176 Clarification – Remplacer <terminator/action statement> par <terminator/action> – Paragraphe 11.13.5: Décision, Grammaire textuelle concrète

Dans le premier alinéa après la syntaxe, remplacer "<terminator statement>" par "<terminator>" et "<action statement>" par <action> (se référer à la modification 4.155).

4.177 Correction d'erreur – Evaluation de réponses de décision – Paragraphe 11.13.5: Décision

Remplacer la phrase suivante dans le premier alinéa du paragraphe *Sémantique*:

"Une décision transfère l'interprétation vers le chemin sortant dont la condition d'intervalle contient le résultat donné par l'interprétation de la question."

par:

"Une décision transfère l'interprétation vers le chemin sortant dont la condition *Range-condition* contient le résultat donné par l'interprétation de la question. La décision de l'appartenance de la question *Decision-question* à chaque réponse *Decision-answer* est faite une seule fois pour chacune des réponses *Decision-answer* dans un ordre quelconque, jusqu'à ce qu'une condition *Range-condition* contenant la question *Decision-question* soit rencontrée, ou jusqu'au moment où cette décision nécessite soit l'interprétation d'une application d'opération qui active une opération, soit le choix d'un texte *Informal-text*."

4.178 Erreur d'écriture – Syntaxe erronée – Paragraphe 11.13.5: Décision, Modèle

Modifier comme suit la spécification du syntype défini de manière anonyme:

```
syntype data_type_N =  
  <<package Predefined>>Integer constants 1:N  
endsyntype;
```

4.179 Extension – Autoriser dans les procédures, etc. – Paragraphe 11.14: Liste d'instructions, Grammaire textuelle concrète

Se référer également aux paragraphes 11.12.2.3 et 11.14.2. Autoriser des arrêts dans des procédures, etc.

Ajouter une variante <stop statement> à la règle <terminating statement>.

4.180 Erreur d'écriture – Aligner avec l'Annexe F – Paragraphe 11.14.1: Instruction composée, Grammaire abstraite

Ajouter un nœud *Exception-handler-node* optionnel au nœud *Compound-node*:

```
Compound-node      ::      Connector-name
                        Variable-definition-set
                        [ Exception-handler-node ]
                        Init-graph-node*
                        Transition
                        Step-graph-node*
```

4.181 Erreur d'écriture – Modèle incorrect – Paragraphe 11.14.1: Instruction composée, Modèle

Le *Modèle* n'était pas cohérent avec la sémantique directe de la règle <compound statement>.

Supprimer la Note:

"NOTE – La liste <statement list> non vide transformée devient une liste d'instructions <action statement> et d'instructions <terminator statement> séparées par des points virgules et se terminant par un point virgule et peut par conséquent être traitée comme une <transition>."

et remplacer la phrase:

"Si la liste <statement list> est vide, le résultat de sa transformation est le texte vide."

par la Note:

"NOTE – Si la liste <statement list> est vide, elle est alors représentée par un nœud *Break-node* comme indiqué dans la *Grammaire graphique concrète*."

4.182 Correction d'erreur – Autoriser des procédures distantes – Paragraphe 11.14.2: Actions et terminateurs de transition en tant qu'instructions, Grammaire textuelle concrète

Les appels de procédure distante ont été omis par erreur. Modifier comme suit la règle <call statement>:

```
<call statement> ::=
    call { <procedure call body> | <remote procedure call body> }
```

4.183 Clarification – Remplacer <action statement> par <action> – Paragraphe 11.14.2: Actions et terminateurs de transition en tant qu'instructions

Dans le premier paragraphe, remplacer "<action statement>" par "<action>" (se référer à la modification 4.155).

4.184 Erreur d'écriture – Traitement d'exception – Paragraphe 11.14.2: Actions et terminateurs de transition en tant qu'instructions transition, Grammaire textuelle concrète

Modifier les règles suivantes comme indiqué ci-dessous:

```
<algorithm action statement> ::=
    output <output body> <end>
    | create <create body> <end>
    | set <set body> <end>
    | reset <reset body> <end>
    | export <export body> <end>
<return statement> ::=
    return <return body> <end>
```

<raise statement> ::=
 raise <raise body> <end>

4.185 Extension – Autoriser des arrêts dans les procédures, etc. – Paragraphe 11.14.2: Actions et terminateurs de transition en tant qu'instructions

Se référer également aux paragraphes 11.12.2.3 et 11.14. Autoriser des arrêts dans les procédures, etc.

Dans la *Grammaire textuelle concrète*, ajouter la production suivante à la règle <stop statement>:

<stop statement> ::=
 stop <end>

Modifier comme suit le dernier alinéa du paragraphe *Modèle* débutant par "La transformation ...":

"La transformation d'une instruction <algorithm action statement>, <return statement>, <stop statement> et <raise statement> est obtenue en supprimant l'élément <end> final.

4.186 Clarification – Paragraphe 11.14.4: Instruction Si, Modèle

Simplification d'utilisation du *Modèle* pour l'instruction <decision statement>.

Modifier:

"L'instruction <if statement> équivaut à l'instruction <action statement> suivante, qui indique une décision <decision>:

```
decision <Boolean expression> ;  
    ( true ) : task { <consequence statement>-transform } ;  
    ( false ) : task { <alternative statement>-transform } ;  
enddecision ;
```

La transformation de l'instruction <alternative statement> est insérée seulement si l'instruction <alternative statement> était présente."

par:

"L'instruction <if statement> équivaut à l'instruction <decision statement> suivante:

```
decision ( <Boolean expression> ) {  
    ( true ) : <consequence statement>  
    ( false ) : <alternative statement>  
}
```

Si l'instruction <alternative statement> n'était pas présente, une instruction <empty statement> est alors insérée à sa place."

4.187 Erreur d'écriture – Paragraphe 11.14.5: Instruction de décision

La variante **any** a été omise par erreur, la syntaxe doit être la suivante:

<decision statement> ::=
 decision ({ <expression> | **any** }) [<comment body>] <left curly bracket>
 <decision statement body>
 <right curly bracket>

4.188 Erreur d'écriture – Mot clé – Paragraphe 11.14.6: Instruction de boucle, Grammaire textuelle concrète

Se référer également aux paragraphes 5.1, 6.1 et D.2.1.

Il convient d'utiliser le mot clé **loop**.

Remplacer "**for**" par "**loop**" dans l'instruction <loop statement>.

4.189 Erreur d'écriture – Paragraphe 11.14.9: Instruction d'exception, Grammaire textuelle concrète

Remplacer "*Local-scope-node*" par "*Compound-node*" (référence incorrecte)

4.190 Erreur d'écriture – Traitement d'exception – Paragraphe 11.15: Temporisateur, Grammaire textuelle concrète

Remplacer les règles <reset> et <set> par les quatre règles suivantes:

```
<reset> ::=
    reset <reset body> [ <end> <on exception> ]

<reset body> ::=
    ( <reset clause> { , <reset clause> }* )

<set> ::=
    set <set body> [ <end> <on exception> ]

<set body> ::=
    <set clause> { , <set clause> }*
```

4.191 Erreur d'écriture – Traitement d'exception – Paragraphe 11.16.2: On-exception, Grammaire textuelle concrète

Supprimer l'élément "<end>" dans la règle <on exception> de manière à obtenir:

```
<on exception> ::=
    onexception <exception handler name>
```

4.192 Clarification – Remplacer <action statement> par <action> – Paragraphe 11.16.3: Traitement, Modèle

Dans l'avant-dernier alinéa, remplacer "<action statement>" par "<action>" (se référer à la modification 4.155).

4.193 Erreur d'écriture – Parties absentes, aligner avec l'Annexe F – Paragraphe 12.1: Définition de données, Grammaire abstraite

Modifier les règles suivantes en ajoutant les ensembles "*Data-type-definition-set*", "*Syntype-type-definition-set*" et "*Exception-definition-set*":

```
Value-data-type-definition      ::      Sort
                                         Data-type-identifier
                                         Literal-signature-set
                                         Static-operation-signature-set
                                         Dynamic-operation-signature-set
                                         Data-type-definition-set
                                         Syntype-definition-set
                                         Exception-definition-set

Object-data-type-definition     ::      Sort
                                         Data-type-identifier
                                         Literal-signature-set
                                         Static-operation-signature-set
                                         Data-type-definition-set
                                         Syntype-definition-set
                                         Exception-definition-set

Interface-definition           ::      Sort
                                         Data-type-identifier*
                                         Data-type-definition-set
                                         Syntype-definition-set
```

Modifier les règles suivantes pour faire la distinction entre les sortes étendues et référencées:

<i>Sort-reference-identifieur</i>	=	<i>Sort-identifieur</i>
		<i>Syntype-identifieur</i>
		<i>Expanded-sort</i>
		<i>Reference-sort</i>
<i>Sort-identifieur</i>	=	<i>Identifieur</i>
<i>Expanded-sort</i>	::	<i>Sort-identifieur</i>
<i>Reference-sort</i>	::	<i>Sort-identifieur</i>

4.194 Extension – Types de donnée en ligne – Paragraphe 12.1: Définitions de données, Grammaire textuelle concrète

Remplacer les règles <sort> et <basic sort> par:

```

<sort> ::=
    <basic sort> [ ( <range condition> ) ]
    |
    <anchored sort>
    |
    <expanded sort>
    |
    <reference sort>
    |
    <pid sort>
    |
    <inline data type definition>
    |
    <inline syntype definition>

<inline data type definition> ::=
    { value | object } [<data type specialization>]
    [ <left curly bracket> <data type definition body> <right curly bracket> ]
    |
    { value | object } [<data type specialization>]
    <end> <data type definition body> <data type closing> <end>

<inline syntype definition> ::=
    syntype <basic sort>
    [ <left curly bracket>
        { <default initialization> [ [<end>] <constraint> ] | <constraint> } <end>
    <right curly bracket> ]
    |
    syntype <basic sort>
    { <default initialization> [ [<end>] <constraint> ] | <constraint> } <end>
    <syntype closing> <end>

<basic sort> ::=
    <datatype type expression>
    |
    <syntype>

```

4.195 Correction d'erreur – Valeur ou objet pour une sorte ancrée – Paragraphe 12.1: Définitions de données, Grammaire textuelle concrète

Il n'est pas permis de spécifier qu'une sorte <anchored sort> est toujours obtenue par héritage sous forme de valeur ou de variante d'objet de la sorte du type de données. Cette interdiction est facilitée par les modifications suivantes:

```

<expanded sort> ::=
    value { <basic sort> | <anchored sort> }

<reference sort> ::=
    object { <basic sort> | <anchored sort> }

```

4.196 Extension – Types de donnée en ligne – Paragraphe 12.1: Définitions de données, Modèle

Ajouter les alinéas suivants à la fin du paragraphe.

"Une sorte <sort> qui est une sorte <basic sort> avec une condition <range condition> est une syntaxe concrète dérivée pour un type <syntype> d'une définition <syntype definition> implicite qui

possède un nom anonyme. Les éléments de cette définition <syntype definition> anonyme sont limités par la condition <range condition> si la sorte <basic sort> a été construite au moyen du constructeur de type de données littéral; dans le cas contraire la condition <range condition> fait partie de la contrainte <size constraint>.

Une définition <inline data type definition> est une syntaxe concrète dérivée pour une sorte <basic sort> d'une définition <data type definition> implicite possédant un nom anonyme. Cette règle <data type definition> anonyme est dérivée de la définition <inline data type definition> en insérant le **type** et le nom anonyme après le mot clé **value** ou **object** dans la définition <inline data type definition>. Chaque définition <inline data type definition> définit une règle <data type definition> implicite différente.

Une définition <inline syntype definition> est une syntaxe concrète dérivée pour une sorte <basic sort> d'une définition <syntype definition> implicite possédant un nom anonyme. Cette définition <syntype definition> anonyme est dérivée de la règle <inline syntype definition> en insérant le nom anonyme et le signe <equals sign> après le type syntype dans la définition <inline syntype definition>."

4.197 Clarification – Paragraphe 12.1.1: Définition de type de données

Insérer le texte suivant immédiatement au début du paragraphe 12.1.1 afin de clarifier la signification de la construction du type de données en ce qui concerne les valeurs littérales:

"Une définition de type de données possède un corps qui contient en général un constructeur de type de données et une indication si le type de données concerne une **valeur** ou un **objet**.

Le constructeur de type de données définit la manière de construire des ensembles de valeurs (valeurs structurées, valeurs littérales et valeurs de choix). Si la définition de type de données est un type **value**, ces valeurs sont alors des éléments de la sorte. Si la définition de type de données est un type **object**, ces valeurs constituent la cible des références faites par les éléments de la sorte."

4.198 Erreur de syntaxe – Élément <end> – Paragraphe 12.1.1: Définitions des types de données, Grammaire textuelle concrète

Se référer également aux paragraphes 12.1.9.4 et 12.3.3.2.

Modifier comme suit la règle <data type definition body>:

```
<data type definition body> ::=  
    {<entity in data type>}* [<data type constructor>] <operations>  
    [<default initialization> <end> ]
```

4.199 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 12.1.1: Définitions des types de données, Grammaire textuelle concrète

Modifier comme suit la règle <data type reference>:

```
<data type reference> ::=  
    <type preamble>  
    { value | object } type <type reference heading> <type reference properties>
```

en la faisant suivre du nouvel alinéa:

"Un en-tête <type reference heading> qui fait partie d'une référence <data type reference> doit posséder un nom <data type name>."

4.200 Clarification – Paragraphe 12.1.1: Définitions des types de données, Grammaire textuelle concrète

Ajouter à la fin:

"Pour chaque signature <operation signature> d'un ensemble <operation signatures> il existera une et une seule définition correspondante (<operation definition> ou <textual operation reference> ou <external operation definition>) dans les définitions <operation definitions> des <operations>."

4.201 Clarification – Paragraphe 12.1.1: Définitions des types de données, Grammaire graphique concrète

Remplacer le texte "<graphical type reference heading> qui contient" par le texte "<type reference heading> avec" de manière à obtenir l'alinéa révisé suivant:

"La zone <type reference area> qui fait partie d'une zone <data type reference area> doit avoir un en-tête <type reference heading> avec un nom <data type name>".

4.202 Erreur d'écriture – Barre verticale superflue – Paragraphe 12.1.2: Définition d'interface, Grammaire textuelle concrète

Supprimer les barres verticales pour les variantes avant <virtuality> dans le texte suivant:

```
|    {<package use clause>}*  
|    [<virtuality>] <interface heading>  
|    [<interface specialization>] <end>  
|    {<package use clause>}*  
|    [<virtuality>] <interface heading>
```

de manière à obtenir le texte:

```
|    {<package use clause>}*  
|    [<virtuality>] <interface heading>  
|    [<interface specialization>] <end>  
|    {<package use clause>}*  
|    [<virtuality>] <interface heading>
```

4.203 Erreur d'écriture – Paramètres de contexte absents – Paragraphe 12.1.2: Définition d'interface, Grammaire textuelle concrète

Modifier comme suit la règle <interface reference>:

```
<interface reference> ::=  
    [<virtuality>]  
    interface <type reference heading> <type reference properties>
```

en la faisant suivre du nouvel alinéa:

"Un en-tête <type reference heading> qui fait partie d'une règle <interface reference> doit posséder un nom <interface name>."

4.204 Clarification – Paragraphe 12.1.2: Définition d'interface, Grammaire graphique concrète

Remplacer le texte "<graphical type reference heading> qui contient" par le texte "<type reference heading> avec" de manière à obtenir l'alinéa révisé suivant:

La zone <type reference area> qui fait partie d'une zone <interface reference area> doit avoir un en-tête <type reference heading> avec un nom <interface name>.

4.205 Clarification – Interfaces – Paragraphe 12.1.2: Définition d'interface, Sémantique

Il est nécessaire que le contenu d'une interface soit défini pour définir l'héritage d'interface.

Insérer le nouvel alinéa suivant après de premier alinéa du paragraphe *Sémantique*:

"Le contenu d'une interface est l'ensemble de tous les signaux, procédures distantes et variables distantes qui sont définis dans une entité <entity in interface> de l'interface, ou qui font l'objet d'une référence dans la liste <interface use list> ou qui sont contenus dans l'interface par le biais d'une spécialisation (c'est-à-dire, par héritage ou par paramétrage du contexte)."

Toutefois, dans le paragraphe 12.1.2, la définition d'interface ne doit pas contenir de paragraphe *Sémantique* à cet endroit du fait qu'il n'y a pas de paragraphe *Grammaire abstraite* à cet endroit. Il convient donc de remplacer le titre "*Sémantique*" par "*Modèle*" et de supprimer ce dernier.

4.206 Clarification – Interface implicite – Paragraphe 12.1.2: Définition d'interface, Modèle

Se référer également au paragraphe 12.1.6.

Le modèle n'était pas correct. Remplacer le texte du paragraphe *Modèle* par:

"Les interfaces sont définies de manière implicite par des définitions d'agent et de type d'agent ainsi que par les définitions des machines d'état d'agent et de type d'agent. L'interface définie de manière implicite pour un agent ou sur un type d'agent possède le même nom que l'agent ou le type d'agent qui l'a définie ainsi que la même unité de domaine de valeur. L'interface définie de manière implicite pour une machine d'état possède le même nom que l'agent ou le type d'agent qui la contient, mais elle est définie pour la même unité de domaine de valeur que la machine d'état qui l'a définie; c'est-à-dire au sein de l'agent ou du type d'agent.

NOTE 1 – Du fait que tout agent et type d'agent possède une interface définie de manière implicite avec un nom identique, toute interface définie de manière explicite doit posséder un nom différent pour tout agent et type d'agent défini dans le même domaine de validité pour éviter un conflit de noms.

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.

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.

L'interface définie par un agent basé sur un type contient dans sa spécialisation <interface specialization> l'interface définie par son type.

NOTE 2 – Pour éviter des lourdeurs dans le texte, la phrase "la sorte de pid de l'agent A" est utilisée fréquemment par convention à la place de la phrase "la sorte de pid définie par l'interface définie de manière implicite par l'agent A" lorsque aucune confusion n'est possible."

4.207 Erreur d'écriture – Aligner avec l'Annexe F – Paragraphe 12.1.4: Opérations, Grammaire abstraite

Ajouter l'élément *Identifier* à:

Operation-signature :: *Operation-name*
*Formal-argument**
[Result]
Identifier

et ajouter l'alinéa suivant à la suite des règles de grammaire:

"L'élément *Identifier* désigne, dans une signature d'opérateur, un identificateur anonyme de la procédure anonyme correspondant à l'opération."

4.208 Erreur d'écriture – Ambiguïté de syntaxe – Paragraphe 12.1.4: Opérations, Grammaire textuelle concrète

Modifier comme suit la règle <operation preamble>:

<operation preamble> ::=
[<virtuality> [<visibility>] | <visibility> [<virtuality>]]

4.209 Erreur d'écriture – Emplacement du texte – Paragraphe 12.1.5: Quelconque (Any)

Supprimer les deux dernières lignes avant le début de la *Grammaire textuelle concrète*.

Déplacer vers le début du paragraphe *Sémantique* le texte restant du paragraphe explicatif non normatif, à l'exception du premier alinéa.

4.210 Clarification – Interfaces implicites – Paragraphe 12.1.6: Pid et sortes de pid, Sémantique

Se référer également au paragraphe 12.1.2.

Remplacer:

"b) si le signal est contenu dans un accès vers l'intérieur relié à la machine à états de l'agent qui a implicitement défini l'interface;

c) la vérification de compatibilité ..."

par:

b) la vérification de compatibilité ...".

4.211 Erreur d'écriture – Paragraphe 12.1.7.3: Type de données de choix, Modèle

[*Cette erreur n'affecte pas la version française*]

4.212 Erreur d'écriture – Sortes optionnelles – Paragraphe 12.1.8: Comportement des opérations, Grammaire graphique concrète

Supprimer l'alinéa:

"Pour chaque diagramme <operation diagram>, il doit y avoir une signature <operation signature> dans la même unité de portée ... De même, la sorte <sort> peut être omise dans le résultat <operation result>."

4.213 Erreur d'écriture – Paragraphe 12.1.8: Comportement des opérations, Sémantique

Remplacer l'alinéa:

"Une définition <external operation definition> est un opérateur ou une méthode dont le comportement n'est pas inclus dans la description du SDL. De manière conceptuelle, il est supposé qu'un comportement est attribué à une définition <external operation definition> et qu'elle est transformée en une définition <operation definition> faisant partie de la transformation du système générique (voir Annexe F)."

par:

"Une définition <external operation definition> est un opérateur ou une méthode dont le comportement n'est pas inclus dans la description du langage SDL (se référer au paragraphe 13)."

4.214 Extension – Signe de domaine – Paragraphe 12.1.9.1: Classe de nom, Grammaire textuelle concrète

Se référer également aux paragraphes 6.1 et 12.1.9.5.

Modifier comme suit la règle <regular interval>:

```
<regular interval> ::=  
    <character string> { <colon> | <range sign> } <character string>
```

4.215 Erreur de syntaxe – Élément <end> – Paragraphe 12.1.9.4: Syntypes, Grammaire textuelle concrète

Se référer également aux paragraphes 12.1.1 et 12.3.3.2.

Modifier la première variante dans la production pour la règle <syntype definition> et supprimer la barre verticale avant la variante <type preamble> de manière à faire débiter la règle comme suit:

```
<syntype definition> ::=  
    {<package use clause>}*  
    syntype <syntype name> <equals sign> <parent sort identifier>  
        { { <default initialization> [ [<end>] <constraint> ] | <constraint> } <end>  
          <syntype closing>  
          | [<syntype closing>] } <end>  
    | {<package use clause>}*  
      <type preamble> <data type heading> [<data type specialization>]
```

4.216 Erreur d'écriture – Variante omise – Paragraphe 12.1.9.4: Syntypes, Grammaire textuelle concrète

Ajoutez comme suit la variante omise par erreur dans la règle <syntype definition>:

```
    | syntype <syntype name> <equals sign> <parent sort identifier>  
      <left curly bracket>  
        [ { <default initialization> [ [<end>] <constraint> ] | <constraint> } <end> ]  
      <right curly bracket>
```

4.217 Modification – Difficulté grammaticale – Paragraphe 12.1.9.4: Syntypes, Grammaire textuelle concrète

Modifier comme suit la deuxième production pour la règle <syntype definition>:

```
    | {<package use clause>}*  
      <type preamble> <data type heading> [<data type specialization>]  
      {<end> <data type definition body> <constraint> <end> <data type closing> <end>  
        | <left curly bracket> <data type definition body> <constraint> <end>  
        <right curly bracket> }
```

4.218 Erreur d'écriture – Ambiguïté – Paragraphe 12.1.9.4: Syntypes, Grammaire textuelle concrète

Ajouter l'alinéa suivant après la syntaxe à la fin de *Grammaire textuelle concrète*:

"Si un élément <contraint> peut être interprété comme appartenant à l'initialisation <default initialization> ou à la définition <syntype definition>, on considérera alors qu'il fait partie de l'initialisation <default initialization>."

4.219 Extension – Signe de domaine – Paragraphe 12.1.9.5: Contrainte, Grammaire textuelle concrète

Se référer également aux paragraphes 6.1 et 12.1.9.1.

Modifier comme suit la règle <closed range>:

$$\text{<closed range>} ::=$$
$$\langle \text{constant} \rangle \{ \langle \text{colon} \rangle \mid \langle \text{range sign} \rangle \} \langle \text{constant} \rangle$$

4.220 Erreur d'écriture – Paragraphe 12.1.9.6: Définition de synonyme, Grammaire textuelle concrète

Remplacer l'alinéa:

"Un élément <external synonym definition item> définit un synonyme <synonym> dont le résultat n'est pas défini dans une spécification. De manière conceptuelle, il est supposé qu'un résultat est attribué à un élément <external synonym definition item> et que celui-ci est transformé en un élément <internal synonym definition item> faisant partie de la transformation du système générique (voir Annexe F)."

par:

"Un élément <external synonym definition item> définit une règle <synonym> dont le résultat n'est pas défini dans une spécification (se référer au paragraphe 13)."

4.221 Erreur d'écriture – Paragraphe 12.2.1: Expressions, Grammaire abstraite

Ajouter l'expression "*State-expression*" comme variante de l'expression "*Active-expression*".

4.222 Extension – Types de donnée en ligne – Paragraphe 12.2.1: Expressions, Grammaire textuelle concrète

Remplacer la règle `<expression>` par les deux règles suivantes:

$$\langle \text{expression} \rangle ::=$$

```

    <expression0>
|   <range check expression>

```

$$\langle \text{expression0} \rangle ::=$$

```

    <operand>
|   <create expression>
|   <value returning procedure call>

```

Supprimer l'élément "<range check expression>" dans la règle <operand 2> de manière à obtenir la règle:

```

<operand2> ::=
    <operand3>
    | <operand2> { <greater than sign>
                  | <greater than or equals sign>
                  | <less than sign>
                  | <less than or equals sign>
                  | in } <operand3>
    | <equality expression>

```

Modifier comme suit la règle "<constant expression>":

```

<constant expression> ::=
    <constant expression0>

```

Remplacer "Une expression <expression> qui ..." par "Une expression <expression0> qui ...".

4.223 Erreur d'écriture – Paragraphe 12.2.1: Expressions, Grammaire textuelle concrète

Dans le premier alinéa, modifier:

"Une expression <expression> qui ne contient aucun primaire <active primary>, aucune expression <create expression> ni aucun appel <value returning procedure call> avec un corps <remote procedure call body> est une expression <constant expression>. Une expression <constant expression> représente une expression *Constant-expression* dans la syntaxe abstraite."

de manière à obtenir les soulignements corrects comme suit:

"Une expression <expression> qui ne contient aucun primaire <active primary>, aucune expression <create expression> ni aucun appel <value returning procedure call> avec un corps <remote procedure call body> est une expression <constant expression>. Une expression <constant expression> représente une expression *Constant-expression* dans la syntaxe abstraite."

Souligner en outre "constant" dans le deuxième alinéa comme suit:

"Une expression <expression> qui n'est pas une <constant expression> représente une expression *Active-expression*."

4.224 Erreur d'écriture – Phrase fausse – Paragraphe 12.2.7: Application d'opération, Grammaire textuelle concrète

Supprimer l'alinéa "Si **this** est utilisé, l'identificateur <operation identifier> doit désigner un opérateur ou une méthode englobante."

4.225 Clarification – Traitement de paramètre – Paragraphe 12.2.7: Application d'opération, Sémantique

Supprimer l'alinéa débutant par "Si une <expression> dans <actual parameters> est omise, ...".

4.226 Correction d'erreur – Restriction de paramètres – Paragraphe 12.2.7: Application d'opération, Modèle

Le texte suivant doit être ajouté au paragraphe *Modèle* de manière à autoriser l'application d'une méthode à un primaire qui n'est pas une variable:

"Si l'élément <primary> d'une application <method application> n'est pas une variable ou le mot clé **this**, ceci constitue une attribution implicite de cet élément à une variable implicite avec la sorte du premier paramètre de l'opération (c'est-à-dire, la sorte de la méthode). L'attribution est placée avant l'action dans laquelle apparaît l'application <method application>. La variable implicite remplace l'élément <primary> dans l'application <method application>."

4.227 Correction d'erreur – Règles trop restrictives pour l'initialisation des données – Paragraphe 12.3.1: Définition de variable

Les règles d'initialisation étaient trop strictes et empêchaient une initialisation raisonnable des objets.

La condition suivante dans la *Grammaire abstraite*:

"Si l'expression *Constant-expression* est présente ..."

est remplacée par:

"Si l'expression *Constant-expression* est présente, elle doit alors avoir l'une des sortes suivantes:

- 1) la même sorte que l'identificateur Sort-reference-identifier désigné, ou
- 2) si la sorte désignée est une sorte d'objet OS, la sorte indiquée par value OS, ou
- 3) si la sorte désignée est une sorte de valeur VS, la sorte indiquée par object OS."

Modifier comme suit le premier alinéa du paragraphe *Sémantique*:

"Lorsqu'une variable est créée et que l'expression *Constant-expression* est présente, la variable est associée avec:

- 1) le résultat de l'expression *Constant-expression*, si la sorte de la variable et l'expression *Constant-expression* sont identiques;
- 2) un objet (distinct de tout autre objet) qui fait référence à l'expression *Constant-expression* si la variable possède une sorte d'objet et si l'expression *Constant-expression* possède une sorte de valeur;
- 3) la valeur à laquelle fait référence l'expression *Constant-expression*, si la variable possède une sorte de valeur et si l'expression *Constant-expression* possède une sorte d'objet.

Dans le cas contraire, si aucune expression *Constant-expression* s'applique, aucun élément de données n'est associé à la variable, c'est-à-dire que cette dernière est "indéfinie".

4.228 Erreur de syntaxe – Élément <end> – Paragraphe 12.3.3.2: Initialisation par défaut, Grammaire textuelle concrète

Se référer également aux paragraphes 12.1.9.4 et 12.1.1.

Modifier comme suit la règle <default initialization>:

<default initialization> ::=
 default [<virtuality>] [<constant expression>]

4.229 Erreur d'écriture – Paragraphe 12.3.4.6: Expression d'état

Se référer également au paragraphe 12.2.1 *Grammaire abstraite*, Expressions.

Remplacer le contenu de la totalité du paragraphe par le texte suivant:

"Grammaire abstraite

State-expression :: *()*"

Grammaire textuelle concrète

<state expression> ::=
 state

Sémantique

Une expression d'état indique le littéral sous forme de chaîne de caractères qui contient l'énoncé du nom de l'état entré le plus récemment dans l'unité de domaine de validité qui l'enclot directement. S'il n'existe pas un tel état, l'expression <state expression> indique alors une chaîne vide (")."

4.230 Erreur d'écriture – Éléments omis et dupliqués – Paragraphe 13.1: Définition optionnelle, Grammaire textuelle concrète

Ajouter "<agent reference>", "<signal reference>" et "<interface reference>" comme variantes de la production pour la règle <select definition>.

Supprimer la variante pour "<textual typebased state partition definition>" dans la règle <select definition>.

4.231 Clarification – Paragraphe 13.1: Définition optionnelle, Grammaire textuelle concrète

Dans la règle <select definition> remplacer "<state partitioning>" par "<state partition>".

4.232 Clarification – Remplacer <action statement> par <action> – Paragraphe 13.2: Chaîne de transition optionnelle, Modèle

Remplacer "<action statement>" par "<action>" dans les éléments a) et b) (se référer à la modification 4.155).

4.233 Clarification – Remplacer <terminator statement> par <terminator> – Paragraphe 13.2: Chaîne de transition optionnelle, Modèle

Remplacer "<terminator statement>" par "<terminator>" dans l'élément a) (se référer à la modification 4.155).

4.234 Erreur d'écriture – Mot clé de quantification – Paragraphe D.2.1: Axiomes, Grammaire textuelle concrète

Se référer également aux paragraphes 5.1, 6.1 et D.2.1.

Ajouter après la règle <quantification> la Note suivante:

"NOTE – **for** est considéré comme un mot clé SDL pour les besoins de la présente annexe."

4.235 Clarification – Supprimer la quantification optionnelle – Paragraphe D.2.1: Axiomes, Grammaire textuelle concrète

Remplacer:

- "c) un identificateur <value identifier> s'il y a une définition de ce nom dans une <quantification> d'équations <quantified equations> englobant le terme <term>, qui doit alors avoir une sorte adaptée à ce contexte; sinon
- d) un identificateur <value identifier> qui a une équation quantifiée implicite pour l'équation <unquantified equation>.

Deux occurrences au moins du même identificateur de valeur non liée dans la même équation <unquantified equation> (ou si cette équation est contenue dans une équation <conditional equation>, à l'intérieur de cette équation <conditional equation>), indiquent une <quantification>.

Dans une équation <unquantified equation> (ou si cette équation est contenue dans une équation <conditional equation> à l'intérieur de cette équation <conditional equation>), il doit y avoir exactement une sorte pour chaque identificateur de valeur quantifiée implicitement qui soit cohérente avec toutes ses utilisations."

par:

- "c) un identificateur <value identifier> s'il existe une définition de ce nom dans une <quantification> d'équations <quantified equations> englobant le terme <term>, qui doit alors avoir une sorte convenant à ce contexte."

4.236 Clarification – Texte informel non utilisé – Paragraphe D.2.1: Axiomes, Sémantique

Supprimer l'alinéa:

"Si un axiome quelconque contient un texte informel, l'interprétation des expressions n'est pas forcément définie ... n'a pas été donnée en SDL.

4.237 Clarification – Supprimer la quantification optionnelle – Paragraphe D.2.1: Axiomes, Sémantique

Supprimer les phrases:

"La sorte des quantifications implicites est la sorte requise par le ou les contextes d'occurrence de l'identificateur non lié. Si les contextes d'un identificateur de valeur qui a une quantification implicite permettent différentes sortes, l'identificateur est lié à une sorte qui est cohérente avec toutes ses utilisations dans l'équation."

4.238 Clarification – Case des noms d'opérateurs – Paragraphe D.3: Paquetage prédéfini (Predefined)

La case des noms d'opérateur "Make", "Extract", "Modify" doit être modifiée dans la totalité de l'annexe:

c'est-à-dire qu'ils doivent débiter par une majuscule au lieu d'une minuscule.

4.239 Erreur d'écriture – Mot clé inherit absent – Paragraphe D.3.13.2: Utilisation

Insérer le mot clé **inherits** manquant dans la définition:

```
value type Boolset inherits Bag< Boolean > endvalue type Boolset;
```

4.240 Erreur d'écriture – Mots clés en majuscules et minuscules – Appendice III

Effectuer les modifications suivantes dans l'Appendice III:

"a) remplacer tous les mots clés avec le mot clé correspondant <lowercase keyword> (ou <uppercase keyword>);"

en remplaçant "<lowercase keyword>", "lowercase <keyword>" et "<uppercase keyword>" par "uppercase <keyword>" pour obtenir le texte:

"a) remplacer tous les mots clé avec le mot clé correspondant lowercase <keyword> (ou uppercase <keyword>);"

et remplacer l'alinéa d) par:

"d) si un ou plusieurs noms <name> entrent en conflit avec des mots clés <keyword> débutant par une minuscule, remplacer alors le premier caractère par un caractère majuscule."

4.241 Clarifications – Appendice III

Dans l'alinéa 1 b) remplacer "<national>" par "national".

4.242 Clarifications – Appendice III

Remplacer "celle des transitions dupliquées" par "une des transitions dupliquées".

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