



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Z.106

(10/96)

SÉRIE Z: LANGAGES DE PROGRAMMATION

Langage de description et de spécification (SDL)

**Format d'échange commun pour le langage de
description et de spécification**

Recommandation UIT-T Z.106

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE Z
LANGAGES DE PROGRAMMATION

Langage de description et de spécification (SDL)	Z.100–Z.109
Critères d'utilisation et d'applicabilité des techniques de description formelle	Z.110–Z.199
Langage de haut niveau de l'UIT-T (CHILL)	Z.200–Z.299
LANGAGE HOMME-MACHINE	Z.300–Z.499
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.399
Divers	Z.400–Z.499

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

AVANT-PROPOS

L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'Union internationale des télécommunications (UIT). Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études à traiter par les Commissions d'études de l'UIT-T lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la CMNT (Helsinki, 1^{er}-12 mars 1993).

La Recommandation UIT-T Z.106, que l'on doit à la Commission d'études 10 (1993-1996) de l'UIT-T, a été approuvée par la CMNT (Genève, 9-18 octobre 1996).

NOTE

Dans la présente Recommandation, l'expression "Administration" est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue de télécommunications.

© UIT 1997

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

TABLE DES MATIÈRES

	<i>Page</i>
1 Objectifs	1
2 Références normatives	1
3 Abréviations	1
4 Conventions	1
5 CIF niveau 1 (CIF/PR).....	2
5.1 Principes généraux.....	3
5.2 Unités transférables des spécifications SDL.....	3
5.3 Syntaxe CIF/PR	3
5.3.1 Fichier CIF	3
5.3.2 Appel de macro	4
5.4 Exemples.....	4
6 CIF niveau 2 (CIF/GR)	4
6.1 Principes généraux.....	4
6.2 Principes de la représentation graphique	4
6.2.1 Système de coordonnées	4
6.2.2 Directives facultatives de positionnement de texte	5
6.2.3 Directives facultatives pour les lignes de flux	6
6.2.4 Imbrication de diagrammes.....	6
6.2.5 Déclarations obligatoires et complémentaires de diagramme.....	7
6.3 Règles lexicales du CIF/GR.....	7
6.3.1 Directives CIF	7
6.3.2 Caractères "Nouvelle ligne" et "Espace"	7
6.3.3 Détails sur l’affichage de texte.....	8
6.4 Syntaxe CIF/GR.....	8
6.4.1 Règles CIF de type A.....	8
6.4.2 Règles CIF de type B	39
6.5 Commentaires CIF spécifiques aux outils	44
7 Exemples.....	44
7.1 Système du jeu DemonGame.....	44
7.1.1 Système du jeu DemonGame en SDL-GR.....	44
7.1.2 Bloc DemonBlock en SDL-GR.....	45
7.1.3 Processus Demon en SDL-GR.....	46
7.2 Constructions SDL complexes.....	47
7.2.1 Jonction de lignes de flux, exemple 1	47
7.2.2 Jonction de lignes de flux, exemple 2	48
7.2.3 Jonction de lignes de flux, exemple 3	48
7.2.4 Lignes et rectangles englobants	49
7.2.5 Lignes de flux de réponse placées après une décision	50
7.2.6 Information de connexion en commentaire texte et en symboles rapportés sur le diagramme.....	50
7.2.7 Extension de texte	51
7.2.8 Diagramme de macro	52

	<i>Page</i>
7.2.9 Position du texte pour des références d'accès	52
7.2.10 Diagrammes imbriqués	53
7.2.11 Diagramme en plusieurs pages	54
7.2.12 Bloc dans un bloc.....	55
7.3 Situations non traitées par le CIF.....	56
8 Critères de conformité au CIF	57
8.1 Lecture d'un fichier CIF	57
8.2 Tracé automatique	57
8.3 Conservation et utilisation des informations spécifiques aux outils	57
Index des mots clés du CIF	58
Appendice I – Commentaires CIF spécifiques aux outils.....	60
I.1 Maintenance du CIF	60
I.2 Commentaires CIF spécifiques aux outils actuellement disponibles.....	60
I.2.1 Position des commentaires CIF spécifiques aux outils	60

RÉSUMÉ

La présente Recommandation définit le format d'échange commun (CIF, *common interchange format*) pour le langage de description et de spécification (SDL, *specification and description language*), (Recommandation Z.100 de l'UIT-T). Le CIF a été défini pour permettre l'échange de spécifications SDL graphiques (SDL-GR) entre outils de fournisseurs différents n'utilisant pas le même format de stockage. Actuellement, la représentation textuelle de SDL (SDL-PR) est utilisée pour échanger des spécifications, avec le défaut que, l'information graphique étant ignorée, une même spécification est souvent très différente d'un outil à l'autre. Avec le CIF, ce défaut est quasiment éliminé car ce format contient presque toutes les informations graphiques. Le CIF rendra les utilisateurs plus indépendants des fournisseurs d'outils et permettra aux organismes de standardisation d'approuver des spécifications au format SDL-CIF quel que soit l'outil qu'ils utilisent en interne. Il permettra aussi d'augmenter la productivité, chacun utilisant son outil préféré, l'échange de spécifications restant possible entre les divers utilisateurs. Tous les fournisseurs d'outils SDL sont encouragés pour offrir des facilités d'importation et d'exportation de SDL-CIF.

La présente Recommandation détaille comment des descriptions SDL doivent être stockées afin d'être échangeables entre outils de fournisseurs différents. Elle ne prend pas en compte le langage MSC (Z.120). Le SDL-CIF a été conçu comme une extension du SDL-PR et est basé sur la syntaxe SDL-PR; il peut être lu et écrit par des outils comme par des utilisateurs. Toutes les constructions disponibles en SDL-PR sont aussi disponibles en SDL-CIF, à l'exception des appels de macro. En conséquence, la plupart des descriptions SDL-PR existantes sont des descriptions légales en SDL-CIF. Le CIF est un format de stockage ouvert grâce à un mécanisme de déclaration de directives spécifiques aux outils. Ce mécanisme permet à un outil conforme au CIF d'étendre le format standard en ajoutant des informations spécifiques. Le SDL-CIF est aussi facilement implémentable et propose aux fabricants d'outils deux niveaux de conformité et les concepts de directive obligatoire et de directive facultative.

La présente Recommandation introduit d'abord le SDL-CIF. Deux niveaux de conformité sont définis, le premier au niveau du SDL-PR et le second incluant l'information graphique. Puis la grammaire est complètement décrite avec sa sémantique associée. Les directives obligatoires et facultatives sont décrites, ainsi que le format des directives spécifiques aux outils. Les directives spécifiques aux outils actuellement disponibles sont décrites en Appendice I.

Les deux niveaux de conformité au CIF sont définis en tant que niveau 1 et niveau 2. Le niveau 1 est très proche du SDL-PR, mais il supporte des spécifications SDL incomplètes. Le niveau 2 inclut le niveau 1; il supporte presque toutes les informations graphiques des diagrammes SDL-GR. Une spécification CIF doit préciser avec quel niveau elle est conforme. De même, les fabricants d'outils implémentant le CIF doivent aussi préciser quel niveau est pris en compte dans les fonctions d'importation et d'exportation de leurs produits.

CONTEXTE

Depuis un certain nombre d'années, l'utilisation du langage de description et de spécification (SDL) a augmenté, tant dans l'industrie que dans les organismes de normalisation pour la production de standards et de Recommandations. Alors que dans l'industrie SDL est souvent utilisé dans un environnement basé sur un seul outil, le travail de normalisation requiert plus généralement d'intégrer des spécifications SDL provenant de différentes organisations utilisant des outils variés. Cela est d'ailleurs une exigence fréquente dans les projets internationaux.

Jusqu'à l'élaboration de cette Recommandation, le seul moyen d'échanger des spécifications SDL était d'échanger des documents SDL-PR (la représentation textuelle de SDL), ce qui avait pour conséquence de perdre les informations graphiques. Bien que celles-ci ne soient pas nécessaires d'un strict point de vue formel, les informations graphiques sont particulièrement utiles pour améliorer la lisibilité et faciliter la compréhension de la spécification SDL. En définissant un format d'échange de données, la présente Recommandation satisfait un besoin depuis longtemps exprimé pour échanger des spécifications SDL sans perdre les informations graphiques.

FORMAT D'ÉCHANGE COMMUN POUR LE LANGAGE DE DESCRIPTION ET DE SPÉCIFICATION

(Genève, 1996)

1 Objectifs

La présente Recommandation définit le format d'échange commun (CIF) pour des spécifications écrites dans le langage de description et de spécification (SDL) normalisé par l'UIT-T [1]. Ce format est destiné aux fabricants d'outils pour être utilisé comme format d'importation et d'exportation afin de permettre l'échange de spécifications SDL entre outils de source différente. La version décrite ici ne prend pas en compte les nouveaux éléments du langage SDL en cours de discussion pour la version de 1996 et n'inclut pas les extensions Z.105. Même si le format CIF permet aux utilisateurs d'écrire des spécifications directement en CIF, il n'est guère adapté à cela. Il doit plutôt être généré par des outils à partir de représentations graphiques SDL (SDL-GR, *SDL specification in the graphical representation*).

2 Références normatives

La présente Recommandation se réfère à certaines dispositions des Recommandations UIT-T et textes suivants qui de ce fait en sont partie intégrante. Les versions indiquées étaient en vigueur au moment de la publication de la présente Recommandation. Toute Recommandation ou tout texte étant sujet à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des références normatives suivantes. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée.

[1] Recommandation UIT-T Z.100 (1993), *Langage de description et de spécification SDL du CCITT*.

3 Abréviations

La présente Recommandation utilise les abréviations suivantes:

CIF	format d'échange commun (<i>common interchange format</i>)
SDL	langage de description et de spécification du CCITT (<i>CCITT specification and description language</i>)

4 Conventions

Les conventions suivantes sont utilisées pour décrire les règles syntaxiques:

[X]	X est optionnel.
{ X Y }	X et Y sont considérés comme formant un seul élément.
X *	X est utilisé zéro ou plusieurs fois.
X +	X est utilisé une ou plusieurs fois.
X Y	Soit X est utilisé, soit Y.
<yyy: Ax>	est une référence à une autre règle CIF. yyy est le nom de la règle. Ax est le numéro de la règle.
<yyy: Bx>	est une référence à une autre règle CIF. yyy est le nom de la règle. Bx est le numéro de la règle.
YYY	est un symbole terminal. YYY sont les caractères formant ce symbole terminal. Majuscules et minuscules sont confondues, c'est-à-dire que la chaîne yyy est considérée comme équivalente à la chaîne YYY.
<yyy>	est une construction PR de la Recommandation Z.100. Elle ne fait pas partie à proprement parler de la grammaire SDL-CIF, mais elle est reprise ici afin de faciliter la compréhension de la syntaxe CIF.
<beginning of yyy>	est une partie de construction SDL-PR de la Recommandation Z.100.
<end of yyy>	est une partie de construction SDL-PR de la Recommandation Z.100.
<enhanced yyy>	est une construction SDL-PR de la Recommandation Z.100 incluant des directives CIF.

5.3.2 Appel de macro

Dans le SDL-PR, les appels de macro peuvent "apparaître partout où une unité <lexical unit> est autorisée" (4.2.3/Z.100).

En revanche dans le SDL-CIF, les appels de macro ne sont autorisés qu'à l'intérieur d'un symbole de tâche.

La règle de production d'une instruction <action statement> (voir 2.6.8.1/Z.100) est alors remplacée par:

```
<action statement> ::=  
    [ <label> ] { <action> <end> | <macro call> }
```

5.4 Exemples

Voici un exemple de spécification SDL échangeable en CIF:

```
process p;  
start; stop;  
endprocess;
```

6 CIF niveau 2 (CIF/GR)

6.1 Principes généraux

Le CIF niveau 2 est une extension du CIF niveau 1 avec ce qui est appelé les "directives CIF". Celles-ci permettent de décrire les principales caractéristiques de la représentation graphique des objets SDL.

Les directives CIF sont placées avant l'objet qu'elles caractérisent. L'un des principes de conception de la présente Recommandation a été que: toutes les constructions SDL-PR contenant de l'information devant être extraite par un convertisseur CIF doivent posséder un commentaire CIF placé avant. Cela permet à un outil de lecture du format CIF de rechercher le prochain commentaire CIF, d'extraire les informations nécessaires de la représentation SDL-PR suivante puis de commencer à chercher le commentaire CIF suivant.

Plusieurs directives CIF peuvent être associées au même objet SDL.

La première directive CIF d'un objet SDL décrit en général la représentation graphique de la partie principale de l'objet, alors que les autres directives CIF pour ce même objet décrivent la représentation graphique de sous-parties de l'objet.

Le CIF niveau 2 ne décrit pas tous les détails de la représentation graphique car cela aurait considérablement réduit le nombre d'outils conformes à ce niveau: certains éditeurs SDL favoriseront le tracé manuel de symboles alors que d'autres privilégieront le tracé automatique. Gérer à la fois le tracé manuel et le tracé automatique est un problème complexe difficile à résoudre lorsque l'on développe un outil SDL.

Afin de résoudre ce problème, les directives CIF sont classées en trois catégories: obligatoires, facultatives et spécifiques aux outils.

Les directives obligatoires décrivent les caractéristiques graphiques qui ne peuvent pas être calculées automatiquement ou pour lesquelles le calcul automatique donnerait un résultat probablement loin des attentes de l'utilisateur, comme par exemple le tracé de symboles et de canaux dans les diagrammes d'interconnexion.

Dans la majorité des outils d'édition SDL, ces caractéristiques graphiques sont généralement contrôlées par l'utilisateur.

Les directives facultatives décrivent des caractéristiques graphiques pouvant être calculées automatiquement, comme par exemple le tracé de texte à l'intérieur de symbole. Si une information facultative n'est pas fournie, alors l'outil détermine automatiquement sa représentation graphique.

Les directives spécifiques aux outils décrivent des caractéristiques (graphiques ou non) qui ne sont pas couvertes par les directives obligatoires ou facultatives. Elles permettent aux fabricants d'outils SDL d'ajouter de nouvelles directives CIF dans le format de stockage qui peuvent être ensuite analysées par leurs outils ou d'autres s'ils implémentent ces extensions non normalisées.

6.2 Principes de la représentation graphique

6.2.1 Système de coordonnées

L'unité de mesure est le 1/10 mm. L'origine est le coin supérieur gauche de la page. L'axe positif des x part horizontalement de l'origine vers la droite. L'axe positif des y part verticalement de l'origine vers le bas.

Pages

Les diagrammes peuvent être découpés en plusieurs pages, comme cela est décrit dans 2.2.5/Z.100.

Toutefois les fichiers SDL-CIF ne sont pas structurés en pages, mais en fonction de la syntaxe SDL-PR. La découpe en pages est décrite par des commentaires CIF insérés entre des unités syntaxiques. Une page peut contenir des éléments d'unités syntaxiques non adjacentes.

Les pages sont des zones de dessin indépendantes. Les coordonnées doivent être interprétées en fonction du nom courant de la page.

Classification de l'information

L'information graphique CIF peut être classée en 4 classes:

- information relative aux symboles ayant la forme de lignes, appelés par la suite "lignes", comme par exemple les acheminements de signaux, les canaux, les lignes de flux dans des transitions et les associations;
- information relative aux symboles qui n'ont pas la forme de ligne, appelés par la suite "symboles", comme par exemple, les processus ou les sorties de signaux;
- information relative au texte;
- autre information, comme par exemple celle relative à la découpe en pages.

Représentation graphique de symbole

Toutes les informations relatives aux positions et tailles des symboles sont obligatoires.

La position d'un symbole est généralement définie par les coordonnées du coin supérieur gauche du rectangle englobant. Il y a quelques exceptions à cela, où c'est le coin supérieur droit qui sert de référence (comme par exemple les symboles en miroir d'extension de texte ou de commentaire).

La taille d'un symbole est définie par la largeur et la hauteur du rectangle englobant.

Des informations spécifiques au symbole sont parfois ajoutées. Par exemple, pour les symboles existant en version gauche et en version droite, les mots clés "Gauche" et "Droite" peuvent être ajoutés.

Représentation graphique de texte

La position et la taille d'un texte sont des informations facultatives. Elles définissent le rectangle englobant le texte et non le texte lui-même.

6.2.2 Directives facultatives de positionnement de texte

Définir la position d'un texte n'est pas obligatoire en CIF. Cela signifie qu'un outil n'a pas à spécifier la position des textes lors de la génération du fichier CIF, et inversement, qu'un outil analysant un fichier CIF contenant des informations de positionnement de texte peut ne pas utiliser ces indications. Voici quelques recommandations:

Lors de la lecture d'un fichier CIF, l'outil devra essayer de prendre en compte les informations de positionnement de texte trouvées dans le CIF. Si cela n'est pas possible, les textes seront placés automatiquement.

S'il n'y a pas d'information de positionnement de texte dans le fichier CIF, alors les textes seront placés automatiquement.

Certaines informations de positionnement de texte sont plus importantes que d'autres pour régénérer les diagrammes SDL-GR à l'identique. Les fabricants d'outils devront faire attention à la manière de traiter ces positionnements de texte. Ces informations sont classées en 3 catégories, la première étant constituée des informations qui doivent être traitées en priorité:

- groupe 1: nom de canal, liste de signaux, nom d'acheminement de signaux, nom d'accès et sélection;
- groupe 2: connexion, ligne de flux de réponse, référence d'accès, retour;
- groupe 3: définition de diagramme, nom de page, symbole de système, symbole de bloc, symbole de processus, symbole de service, symbole de procédure, symbole d'opérateur, état, mise en réserve de signal, tâche, armement de temporisateur, réinitialisation de temporisateur, création de processus, appel de procédure, départ de procédure, décision, signal continu, condition de validation, option de transition, branchement, étiquette, appel de macro, accès sortant de macro, entrée, entrée prioritaire, sortie, texte et référence de progiciel.

Représentation graphique de ligne

Le tracé de ligne est défini par une liste de coordonnées: le premier élément spécifie le point de départ de la ligne, les éléments intermédiaires spécifient les points de rupture successifs de la ligne et le dernier élément spécifie le point d'arrivée de la ligne.

Généralement, les points de départ et d'arrivée sont connectés à d'autres symboles.

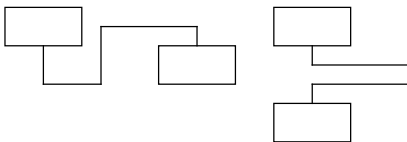
Si un point de départ ou d'arrivée est connecté à un symbole qui n'a pas la forme de rectangle, ce point est spécifié sur le rectangle englobant le symbole et non sur la représentation graphique réelle du symbole, cela pour simplifier les calculs de géométrie.

La représentation graphique d'un canal et d'un acheminement de signaux est une information obligatoire, étant donné qu'il est impossible de deviner ce que l'utilisateur veut réellement.

6.2.3 Directives facultatives pour les lignes de flux

Les lignes de flux de transition ne partant pas d'un symbole de décision sont facultatives, de la même manière que le positionnement de texte. Voici quelques recommandations:

Il est important de fournir des informations lorsque les lignes de flux sont compliquées comme dans les exemples ci-après:



Il est moins important de fournir ces informations lorsque les lignes de flux sont simples comme dans les exemples ci-après:



T1008850-96/d01

Informations graphiques non couvertes par le CIF

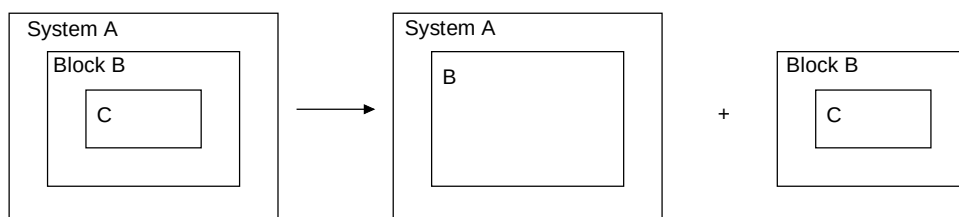
Les directives CIF sont principalement associées au positionnement et à la taille des symboles, car, d'une part, ce sont des informations de base pouvant être implémentées sans problème, et d'autre part, si elles étaient manquantes, les diagrammes seraient très difficiles à retracer.

D'autres types d'information ont été considérés comme moins importants et ne sont pas couverts par le SDL-CIF, comme par exemple, le choix des polices de caractères, la taille des caractères, la couleur ou l'épaisseur des traits.

Ce genre d'information peut très bien être pris en compte par des directives spécifiques aux outils.

6.2.4 Imbrication de diagrammes

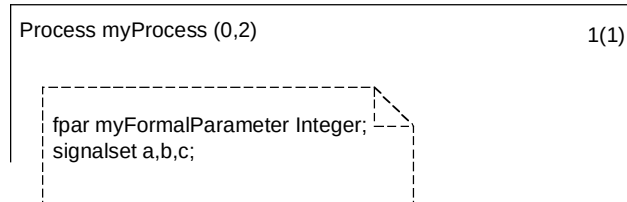
L'imbrication de diagrammes est supportée par le CIF. Un outil ne supportant pas les diagrammes imbriqués peut les convertir en diagrammes séparés au moyen de définitions référencées:



T1008860-96/d02

6.2.5 Déclarations obligatoires et complémentaires de diagramme

Le CIF ne supporte pas les déclarations complémentaires de diagramme. Cela signifie que les outils supportant cette possibilité doivent décomposer le texte de la déclaration de diagramme en deux parties, la première contenant la déclaration obligatoire et la seconde contenant la déclaration complémentaire, et cela sans support du CIF. Notons que le SDL-GR de l'exemple ci-après n'est pas correct, étant donné que, d'après la Recommandation Z.100, il ne doit pas avoir de symbole entourant le texte de la déclaration complémentaire.



T1008870-96/d03

Texte de déclaration de diagramme décomposé en partie obligatoire et partie complémentaire.

6.3 Règles lexicales du CIF/GR

6.3.1 Directives CIF

Les directives CIF sont des formes spéciales de note <note> de la Recommandation Z.100, qui ont toutes en commun la description suivante:

<cif directive> ::=

```
/* { CIF | cif } <text> */
```

Une ligne source dans un fichier CIF analysé, contenant une directive CIF, ne doit pas contenir d'autre marque.

Une note du CIF/GR <note> ne doit pas contenir de directive <cif directive>.

6.3.2 Caractères "Nouvelle ligne" et "Espace"

Les caractères "Nouvelle ligne" et "Espace" sont généralement considérés comme non significatifs lorsqu'ils sont rencontrés dans un fichier CIF, et en conséquence ils sont ignorés.

Toutefois, lorsque deux marques SDL adjacentes sont affichées dans un diagramme en tant que parties adjacentes d'un même objet texte, les caractères "Nouvelle ligne" et "Espace" éventuellement présents entre ces deux marques ne doivent pas être ignorés: ils doivent être considérés comme une partie de l'objet texte.

Cela permet aux outils de prendre en compte la façon dont l'utilisateur a entré le texte.

Lorsque deux marques SDL d'un objet texte sont séparées par des caractères "Espace", puis un caractère "Nouvelle ligne", puis des caractères "Espace", les caractères "Espace" placés avant le premier caractère significatif dans la deuxième ligne doivent être ignorés étant donné qu'ils représentent des espaces d'indentation.

Le premier caractère "Espace" significatif dans une ligne est celui placé au même rang que le premier caractère "/" de la directive CIF en cours d'analyse.

Par exemple, dans l'extrait suivant de SDL-CIF:

```
BLOCK b;
    /* CIF Signalroute (500,400), (300,400) */
    SIGNALROUTE r FROM ENV To P WITH s1 , s2
    s3;
```

Le texte qui doit être affiché pour la liste de signaux est:

```
's1 , s2' // NL // ' s3'
```

(en considérant qu'il n'y a pas de caractères "Espace" après s2).

6.3.3 Détails sur l’affichage de texte

L’insertion de caractères "Nouvelle ligne" dans le SDL-GR doit être préservée dans le SDL-PR. Par exemple, une liste de deux signaux définie dans le GR sur deux lignes doit être stockée en PR ainsi:

```
FROM ENV TO P WITH KeyStroke,  
Card;
```

au lieu de:

```
FROM ENV TO P WITH KeyStroke, Card;
```

6.4 Syntaxe CIF/GR

6.4.1 Règles CIF de type A

Une règle normale (comme par exemple la règle <symbole de départ: A43>) est en général décrite selon la structure suivante:

- une section décrivant la syntaxe du commentaire CIF et l’information associée dans le SDL-PR. Cette section montre aussi comment les commentaires CIF doivent être placés dans le code SDL-PR;
- une section de références croisées indiquant quelles règles utilisent la règle courante;
- une section fournissant des détails sur les constructions SDL-PR prises directement de la Recommandation Z.100: la règle <xxx> est ...;
- une section composée d’un texte explicatif et d’un exemple.

Notons que la grammaire CIF détaillée ici ne fournit que des instructions expliquant comment il faut insérer des commentaires CIF dans du SDL-PR. Un fichier CIF n’est correct que s’il est conforme au SDL-PR.

A1 Description CIF <CIF description: A1>:

```
{ <diagram description: A2> }*
```

Information complémentaire:

Cette règle est la règle de départ d’une description CIF.

A2 Description de diagramme <diagram description: A2>:

```
<diagram start: A3> { <CIF descriptor: A18> }* [<diagram end: A17> ]
```

Cette règle est utilisée par <CIF description: A1>, <CIF descriptor: A18>.

Information complémentaire:

La fin <diagram end: A17> doit être présente si le diagramme n’est pas un diagramme de système de type de base (c’est-à-dire si une définition qui n’est pas du type <textual typebased system definition> est utilisée dans le SDL-PR). Voir aussi <system diagram start: A5>.

A3 Début de diagramme <diagram start: A3>:

```
<package diagram start: A4> | <system diagram start: A5> | <system type diagram start: A6> | <block diagram  
start: A7> | <block type diagram start: A8> | <substructure diagram start: A9> | <process diagram start: A10> | <process  
type diagram start: A11> | <service diagram start: A12> | <service type diagram start: A13> | <procedure diagram  
start: A14> | <operator diagram start: A15> | <macro diagram start: A16>
```

Cette règle est utilisée par <diagram description: A2>.

A4 Début de diagramme de progiciel <package diagram start: A4>:

```
/* CIF PackageDiagram */  
{ <page declaration: B3> }+  
<beginning of package definition>
```

Cette règle est utilisée par <diagram start: A3>.

<beginning of package definition> ::=
{<package reference clause>}* **PACKAGE** <name> [<interface>] <end>

<package reference clause> ::=
USE <package name> [/ <definition selection list>] <end>

<definition selection list> ::=
<definition selection> { , <definition selection> }*

<definition selection> ::=
[<entity kind>] <name>

<entity kind> (en SDL92) ::=
{ **SYSTEM TYPE** } | { **BLOCK TYPE** } | { **PROCESS TYPE** } | { **SERVICE TYPE** } | **SIGNAL** | **PROCEDURE** | **NEWTYPE** | **SIGNALLIST** | **GENERATOR** | **SYNONYM** | **REMOTE**

<entity kind> (en SDL96) ::=
{ **SYSTEM TYPE** } | { **BLOCK TYPE** } | { **PROCESS TYPE** } | { **SERVICE TYPE** } | **SIGNAL** | [**REMOTE**] | **PROCEDURE** | **NEWTYPE** | **SIGNALLIST** | **GENERATOR** | **SYNONYM** | **REMOTE**

<interface> ::=
INTERFACE <definition selection list>

Information complémentaire:

Il y a une déclaration <page declaration: B3> par page du diagramme.

Exemple:



```
/* CIF PackageDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,250),(1700,1950) */
/* CIF PackageReference (125,25) */
use myOtherPackage;
Package myPackage;
```

A5 Début de diagramme de système <system diagram start: A5>:

```
/* CIF SystemDiagram */
{ <page declaration: B3> }+
<enhanced textual system definition>
```

Cette règle est utilisée par <diagram start: A3>.

<enhanced textual system definition> ::=
{<package reference clause>}* { **SYSTEM** <name> <end> |
 <textual typebased system definition> /* CIF End SystemDiagram */ }

Information complémentaire:

Il y a une déclaration <page declaration: B3> par page du diagramme.

Exemple 1:

```

/* CIF SystemDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
SYSTEM mySystem;

```

Exemple 2:

```

/* CIF SystemDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
SYSTEM mySystem : mySystemType;
/* CIF End SystemDiagram */

```

A6 Début de diagramme de type de système <system type diagram start: A6>:

```

/* CIF SystemTypeDiagram */
<diagram parts: B1>
<beginning of system type definition>

```

Cette règle est utilisée par <diagram start: A3>.

<beginning of system type definition> ::=

SYSTEM TYPE { <name> | <identifiant> } [<formal context parameters>] [<specialization>] <end>

<identifiant> ::=

[<qualifier>] <name>

<qualifier> ::=

{ <path item> { / <path item> } * } |
{ << <path item> { / <path item> } * >> }

<path item> (en SDL92) ::=

<scope unit kind> { <name> | <quoted operator> }

<path item> (en SDL96) ::=

<scope unit kind> { <name> | <quoted operator> | <operator name> <exclamation> }

<scope unit kind> ::=

PACKAGE | { **SYSTEM TYPE** } | **SYSTEM** | **BLOCK** | { **BLOCK TYPE** } | **SUBSTRUCTURE** | **PROCESS** |
{ **PROCESS TYPE** } | **SERVICE** | { **SERVICE TYPE** } | **PROCEDURE** | **SIGNAL** | **OPERATOR** | **TYPE**

<quoted operator> ::=

{ <quote> <infix operator> <quote> } | { <quote> <monadic operator> <quote> }

<quote> ::= "

<formal context parameters> ::= ...

<specialization> ::=

INHERITS <type expression> [**ADDING**]

Exemple:

```

System Type <<package myPackage>> mySystemType 1(1)

```

```

/* CIF SystemTypeDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
System Type <<package myPackage>> mySystemType;

```

A7 Début de diagramme de bloc <block diagram start: A7>:

```

/* CIF BlockDiagram */
<diagram parts: B1>
<beginning of block definition>

```

Cette règle est utilisée par <diagram start: A3>.

<beginning of block definition> ::=

BLOCK { <name> | <identifiant> } <end>

Exemple:

Block myBlock	1(1)
---------------	------

```
/* CIF BlockDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Block myBlock;
```

A8 Début de diagramme de type de bloc <block type diagram start: A8>:

```
/* CIF BlockTypeDiagram */
<diagram parts & gate references: B2>
<beginning of block type definition>
```

Cette règle est utilisée par <diagram start: A3>.

```
<beginning of block type definition> ::=
[ <virtuality> ] BLOCK TYPE { <name> | <identifiant> } [ <formal context parameters> ] [ <virtuality constraint> ]
[ <specialization> ] <end>
```

```
<virtuality> ::=
VIRTUAL | REDEFINED | FINALIZED
```

```
<virtuality constraint> ::=
ATLEAST <identifiant>
```

Exemple:

Virtual Block Type myBlockType	1(1)
--------------------------------	------

```
/* CIF BlockTypeDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Virtual Block Type myBlockType;
```

A9 Début de diagramme de sous-structure <substructure diagram start: A9>:

```
/* CIF SubstructureDiagram [ Invisible ] */
[ <diagram parts & gate references: B2> ]
{ <beginning of block substructure definition> |
  <beginning of channel substructure definition> }
```

Cette règle est utilisée par <diagram start: A3>.

```
<beginning of block substructure definition> ::=
SUBSTRUCTURE { <name> | <identifiant> } <end>
```

```
<beginning of channel substructure definition> ::=
SUBSTRUCTURE { <name> | <identifiant> } <end>
```

Information complémentaire:

Le diagramme de sous-structure est **Invisible** s'il n'est pas visible dans le SDL-GR. Cela est le cas lorsque la construction GR de mettre un bloc dans un autre est utilisée.

Si le mot clé **Invisible** est présent, alors la règle <diagram parts & gate references: B2> peut être omise. Si le mot clé **Invisible** est omis, alors la règle <diagram parts & gate references: B2> doit être présente. Pour plus de détails, se reporter au chapitre 7.

Exemple:

Substructure myBlockSubstructure	1(1)
----------------------------------	------

```
/* CIF SubstructureDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Substructure myBlockSubstructure;
```


A10 Début de diagramme de processus <process diagram start: A10>:

/* CIF ProcessDiagram */

<diagram parts: B1>

<beginning of process definition>

Cette règle est utilisée par <diagram start: A3>.

<beginning of process definition> ::=

PROCESS { <name> | <identifier> } [<number of process instances>] <end> [<formal parameters> <end>]
[<valid input signal set>]

<number of process instances> ::=

([<initial number>] [, [<maximum number>]])

<formal parameters> ::=

FPAR <parameters of sort> { , <parameters of sort> }*

<parameters of sort> ::=

<variable name> { , <variable name> }* <sort>

<sort> ::=

<sort identifier> | <syntype>

<syntype> ::=

<syntype identifier>

<valid input signal set> ::=

SIGNALSET [<signal list>] <end>

<signal list> ::=

<signal list item> { , <signal list item> }*

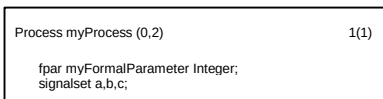
<signal list item> (en SDL92) ::=

<signal identifier> | (<signal list identifier>) | <timer identifier>

<signal list item> (en SDL96) ::=

<signal identifier> | (<signal list identifier>) | <timer identifier> | [**PROCEDURE**] <remote procedure identifier> |
[**REMOTE**] <remote variable identifier>

Exemple:



```
/* CIF ProcessDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Process myProcess(0,2);
fpar myFormalParameter Integer;
signalset a,b,c;
```

A11 Début de diagramme de type de processus <process type diagram start: A11>:

/* CIF ProcessTypeDiagram */

<diagram parts & gate references: B2>

<beginning of process type definition>

Cette règle est utilisée par <diagram start: A3>.

<beginning of process type definition> ::=

[<virtuality>] **PROCESS TYPE** { <name> | <identifier> } [<formal context parameters>] [<virtuality constraint>]
[<specialization>] <end> [<formal parameters> <end>] [<valid input signal set>]

Exemple:

Process Type myProcessType	1(1)
----------------------------	------

```
/* CIF ProcessTypeDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Process Type myProcessType;
```

A12 Début de diagramme de service <service diagram start: A12>:

```
/* CIF ServiceDiagram */
<diagram parts: B1>
<beginning of service definition>
```

Cette règle est utilisée par <diagram start: A3>.

<beginning of service definition> ::=

SERVICE { <name> | <identifier> } <end> [<valid input signal set>]

Exemple:

Service myService	1(1)
-------------------	------

```
/* CIF ServiceDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Service myService;
```

A13 Début de diagramme de type de service <service type diagram start: A13>:

```
/* CIF ServiceTypeDiagram */
<diagram parts: B1>
<beginning of service type definition>
```

Cette règle est utilisée par <diagram start: A3>.

<beginning of service type definition> ::=

[<virtuality>] **SERVICE TYPE** { <name> | <identifier> } [<formal context parameters>] [<virtuality constraint>] [<specialization>] <end> [<valid input signal set>]

Exemple:

Service Type myServiceType	1(1)
inherits myFirstServiceType;	

```
/* CIF ServiceTypeDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Service Type myServiceType inherits myFirstServiceType;
```

A14 Début de diagramme de procédure <procedure diagram start: A14>:

```
/* CIF ProcedureDiagram */
<diagram parts: B1>
<beginning of procedure definition>
```

Cette règle est utilisée par <diagram start: A3>.

<beginning of procedure definition> (en SDL92) ::=

<procedure preamble> **PROCEDURE** { <name> | <identifier> } [<formal context parameters>]
[<virtuality constraint>] [<specialization>] <end> [<procedure formal parameters> <end>]
[<procedure result> <end>]

<beginning of procedure definition> (en SDL96) ::=

<external procedure definition> |
<procedure preamble> **PROCEDURE** { <name> | <identifier> } [<formal context parameters>] [<virtuality constraint>] [<specialization>] <end> [<procedure formal parameters> <end>] [<procedure result> <end>]

<external procedure definition> (en SDL96) ::=

```
PROCEDURE <procedure name> [ <procedure signature> ] EXTERNAL <end>
```

<procedure signature> ::=

```
[ [ <end> ] FPAR <procedure formal parameter constraint> { , <procedure formal parameter constraint> } *
[ <end> RETURNS <sort> ] ] |
[ <end> ] RETURNS <sort>
```

<procedure formal parameter constraint> ::=

```
<parameter kind> <sort>
```

<procedure preamble> ::=

```
[ <virtuality> ] [ EXPORTED [ AS <identifier> ] ]
```

<procedure formal parameters> ::=

```
FPAR <formal variable parameters> { , <formal variable parameters> } *
```

<formal variable parameters> ::=

```
<parameter kind> <parameters of sort>
```

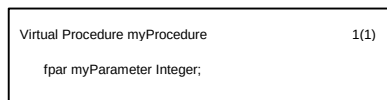
<parameter kind> ::=

```
[ IN/OUT | IN ]
```

<procedure result> ::=

```
RETURNS [ <variable name> ] <sort>
```

Exemple:



```
/* CIF ProcedureDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Virtual Procedure myProcedure;
fpar myParameter Integer;
```

A15 Début de diagramme d'opérateur <operator diagram start: A15>:

```
/* CIF OperatorDiagram */
<diagram parts: B1>
<beginning of operator definition>
```

Cette règle est utilisée par <diagram start: A3>.

<beginning of operator definition> (en SDL92) ::=

```
OPERATOR { <name> | <identifier> } <end> <formal parameters> <end> <operator result> <end>
```

<beginning of operator definition> (en SDL96) ::=

```
<external operator definition> |
OPERATOR { <name> | <identifier> } <end> [ <formal parameters> <end> ] <operator result> <end>
```

<external operator definition> (en SDL96) ::=

```
OPERATOR <operator name> [ <operator signature> ] EXTERNAL <end>
```

<operator signature> ::=

```
<operator name> : <argument list> -> <result> |
ORDERING |
NOEQUALITY
```

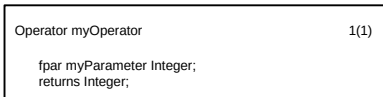
<operator result> (en SDL92) ::=

```
RETURNS [ <variable name> ] <extended sort>
```

<operator result> (en SDL96) ::=

```
RETURNS [ <variable name> ] <sort>
```

Exemple:



```
/* CIF OperatorDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
Operator myOperator;
fpar myParameter Integer;
returns Integer;
```

A16 Début de diagramme de macro <macro diagram start: A16>:

```
/* CIF MacroDiagram */
{ <page declaration: B3> }+
<beginning of macro definition>
```

Cette règle est utilisée par <diagram start: A3>.

```
<beginning of macro definition> ::=
MACRODEFINITION <name> [ <macro formal parameters> ] <end>
```

```
<macro formal parameters> ::=
FPAR <macro formal parameter> { , <macro formal parameter> }*
```

```
<macro formal parameter> ::=
<name>
```

Information complémentaire:

Tous les diagrammes de macro ne sont pas supportés dans le CIF. Les seuls diagrammes supportés correspondent à des macros avec un seul accès entrant et un seul accès sortant et pouvant remplacer un symbole de tâche dans un processus. Voir aussi les règles <macro call symbol: A61>, <macro inlet symbol: A62> et <macro outlet symbol: A63>.

Il y a une déclaration <page declaration: B3> par page du diagramme.

Exemple:

```
/* CIF MacroDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
MACRODEFINITION myMacro;
```

A17 Fin de diagramme <diagram end: A17>:

```
{ /* CIF End PackageDiagram */ <end of package definition> |
/* CIF End SystemDiagram */ <part of end of textual system definition> |
/* CIF End SystemTypeDiagram */ <end of system type definition> |
/* CIF End BlockDiagram */ <end of block definition> |
/* CIF End BlockTypeDiagram */ <end of block type definition> |
/* CIF End SubstructureDiagram */ <end of channel substructure definition> | <end of block substructure definition> |
/* CIF End ProcessDiagram */ <end of process definition> |
/* CIF End ProcessTypeDiagram */ <end of process type definition> |
/* CIF End ServiceDiagram */ <end of service definition> |
/* CIF End ServiceTypeDiagram */ <end of service type definition> |
/* CIF End ProcedureDiagram */ <end of procedure definition> |
/* CIF End OperatorDiagram */ <end of operator definition> |
/* CIF End MacroDiagram */ <end of macro definition> }
```

Cette règle est utilisée par <diagram description: A2>.

<end of package definition> ::=
ENDPACKAGE [<package name>] <end>

<part of end of textual system definition> ::=
ENDSYSTEM [<system name>] <end>

<end of system type definition> ::=
ENDSYSTEM TYPE [<system type name> | <system type identifier>] <end>

<end of block definition> ::=
ENDBLOCK [<block name> | <block identifier>] <end>

<end of block type definition> ::=
ENDBLOCK TYPE [<block type name> | <block type identifier>] <end>

<end of channel substructure definition> ::=
ENDSUBSTRUCTURE [{ <channel substructure name> | <channel substructure identifier> }] <end>

<end of block substructure definition> ::=
ENDSUBSTRUCTURE [{ <block substructure name> | <block substructure identifier> }] <end>

<end of process definition> ::=
ENDPROCESS [<process name> | <process identifier>] <end>

<end of process type definition> ::=
ENDPROCESS TYPE [<process type name> | <process type identifier>] <end>

<end of service definition> ::=
ENDSERVICE [{ <service name> | <service identifier> }] <end>

<end of service type definition> ::=
ENDSERVICE TYPE [{ <service type name> | <service type identifier> }] <end>

<end of procedure definition> ::=
ENDPROCEDURE [<procedure name> | <procedure identifier>] <end>

<end of operator definition> ::=
ENDOPERATOR [{ <operator identifier> | <operator name> }] <end>

<end of macro definition> ::=
ENDMACRO [<macro name>] <end>

Information complémentaire:

Ce commentaire CIF est également utilisé après une définition de type <textual typebased system definition>; voir la règle <system diagram start: A5>.

Exemple:

```

/* CIF End SystemTypeDiagram */
ENDSYSTEM TYPE mySystemType;

```

A18 Descripteur CIF <CIF descriptor: A18>:

<diagram description: A2> | <default size: A19> | <page switch: A20> | <channel: A21> | <signal route: A22> |
 <gate: A23> | <connect: A24> | <create line: A27> | <flow line: A28> | <answer flow line: A29> | <block symbol: A30>
 | <dashed block symbol: A31> | <process symbol: A32> | <dashed process symbol: A33> | <service symbol: A34> |
 <dashed service symbol: A35> | <system type symbol: A36> | <block type symbol: A37> | <process type symbol: A38>
 | <service type symbol: A39> | <block substructure symbol: A40> | <procedure symbol: A41> | <operator symbol: A42>
 | <start symbol: A43> | <stop symbol: A44> | <state symbol: A45> | <nextstate symbol: A46> | <save symbol: A47> |
 <task symbol: A48> | <set symbol: A49> | <reset symbol: A50> | <export symbol: A51> | <create symbol: A52> |
 <procedure call symbol: A53> | <procedure start symbol: A54> | <return symbol: A55> | <decision symbol: A56> |
 <continuous signal symbol: A57> | <enabling condition symbol: A58> | <transition option symbol: A59> | <join
 symbol: A60> | <macro call symbol: A61> | <macro inlet symbol: A62> | <macro outlet symbol: A63> | <label
 symbol: A64> | <input symbol: A65> | <priority input symbol: A66> | <output symbol: A67> | <text symbol: A68> |
 <select symbol: A69> | <descriptor end: A70>

Cette règle est utilisée par <diagram description: A2>.

Information complémentaire:

La règle <diagram description: A2> est utilisée pour décrire des diagrammes imbriqués.

A19 Taille par défaut <default size: A19>:

```
/* CIF DefaultSize <size point: B26> */
```

Cette règle est utilisée par <CIF descriptor: A18>.

Information complémentaire:

Ce commentaire CIF peut être placé avant tout symbole ou ligne de PR présent dans un diagramme.

La taille donnée ici sera utilisée pour tous les symboles qui suivent et qui n'ont pas de taille spécifiée explicitement, jusqu'à ce qu'une nouvelle taille par défaut soit spécifiée. Tant qu'une taille par défaut n'est pas spécifiée, tout symbole doit spécifier sa taille.

Exemple: dans cet exemple, la taille du symbole de tâche sera fixée à (200,100).

```
/* CIF DefaultSize (200,100) */
/* CIF Task (800,550) */
task GameP:=null;
```

A20 Changement de page <page switch: A20>:

```
/* CIF CurrentPage <page name> */
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<page number> ::=
<literal name>
```

Information complémentaire:

Ce commentaire CIF peut être placé avant tout symbole ou ligne de CIF ou de PR présent dans un diagramme. La page mentionnée devient la page courante. Le nom de la page doit se référer à une page déclarée plus haut dans la directive CIF de début de diagramme.

Tout ce qui suit ce commentaire CIF à l'intérieur du diagramme sera placé dans la page courante, jusqu'à ce qu'une nouvelle page soit définie. La page courante du diagramme est définie initialement par la déclaration <page declaration: B3>.

Exemple:

```
/* CIF CurrentPage 1 */
/* CIF Task (800,550) */
task GameP:=null;
```

A21 Canal <channel: A21>:

```
/* CIF Channel <pointlist: B22> [ InvisibleName ] */
[ <channel name text position: B25> ]
[ <first signallist text position: B6> ]
[ <second signallist text position: B7> ]
[ <first arrow position: B8> ]
[ <second arrow position: B9> ]
<enhanced channel definition>
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<enhanced channel definition> (en SDL92) ::=
CHANNEL <name> [ NODELAY ]
<channel path> [ <channel path> ]
[ <channel substructure symbol: B4> <textual channel substructure reference> ]
ENDCHANNEL [ <name> ] <end>
```

<enhanced channel definition> (en SDL96) ::=
CHANNEL [<name>] [**NODELAY**]
 <channel path> [<channel path>]
 [<channel substructure symbol: B4> <textual channel substructure reference>]
ENDCHANNEL [<name>] <end>

<channel path> (en SDL92) ::=
FROM <channel endpoint> **TO** <channel endpoint> **WITH** <signal list> <end>

<channel path> (en SDL96) ::=
FROM <channel endpoint> **TO** <channel endpoint> [**WITH** <signal list>] <end>

<channel endpoint> ::=
 { <block identifier> | **ENV** } [**VIA** <gate name>]

<textual channel substructure reference> ::=
SUBSTRUCTURE <name> **REFERENCED** <end>

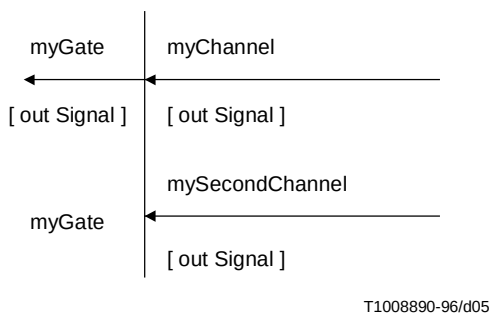
Information complémentaire:

En SDL96 seulement, si le mot clé **InvisibleName** est présent, alors le nom du canal ne doit pas apparaître dans le GR. Le nom est seulement présent dans le PR afin de pouvoir référencer ce canal dans une instruction de connexion **CONNECT**. Le mot clé **InvisibleName** ne doit pas apparaître dans le CIF généré à partir d'une description SDL92.

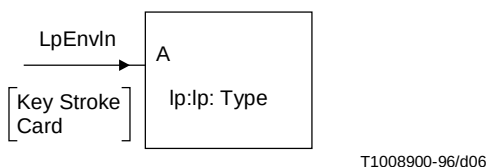
Le premier élément dans la liste des points est situé sur le rectangle englobant le symbole, il correspond à l'instruction **FROM** <channel endpoint> du premier trajet <channel path>. Le dernier élément dans la liste des points est situé sur le rectangle englobant le symbole, il correspond à l'instruction **TO** <channel endpoint> du premier trajet <channel path>.

Si la position <first arrow position: B8> n'est pas fournie, alors la position de la flèche sera calculée automatiquement lors de l'analyse du fichier CIF.

Le positionnement du texte pour l'accès associé à l'instruction **VIA** est spécifié soit par la règle <gate: A23>, soit par la règle <gate reference: B19>. Le schéma ci-dessous montre un accès et une référence à cet accès:



Exemple 1:



```
/* CIF Block (500,350) */
/* CIF GateReference (500,400) */
/* CIF TextPosition (510,390) */
BLOCK lp:lpType;
/* CIF Channel (300,400),(500,400) */
/* CIF TextPosition (390,350) */
/* CIF TextPosition (390,410) SignalList1 */
CHANNEL LpEnvIn
FROM env TO Lp VIA A
WITH KeyStroke,
Card;
ENDCHANNEL LpEnvIn;
```

Exemple 2:

L'exemple ci-dessous présente la sous-structure de canal 'myChannel' connectée au canal 'c'.

```
/* CIF Channel (700,400),(1100,400) */
/* CIF TextPosition (900,390) */
/* CIF TextPosition (700,350) SignalList1 */
/* CIF TextPosition (1100,350) SignalList2 */
CHANNEL c
FROM a TO b WITH s1;
FROM b TO a WITH s2;
/* CIF ChannelSubstructure (800,550) */
/* CIF Line (900,550),(900,400) Dashed */
SUBSTRUCTURE mySubstructure REFERENCED;
ENDCHANNEL c;
```

A22 Acheminement de signal <signal route: A22>:

```
/* CIF SignalRoute <pointlist: B22> [ InvisibleName ] */
[ <signal route name text position: B25> ]
[ <first signallist text position: B6> ]
[ <second signallist text position: B7> ]
<signal route definition>
```

Cette règle est utilisée par <CIF descriptor: A18>.

<signal route definition> (en SDL92) ::=

SIGNALROUTE <name>

<signal route path>

[<signal route path>]

<signal route definition> (en SDL96) ::=

SIGNALROUTE [<name>]

<signal route path>

[<signal route path>]

<signal route path> (en SDL92) ::=

FROM <signal route endpoint> **TO** <signal route endpoint> **WITH** <signal list> <end>

<signal route path> (en SDL96) ::=

FROM <signal route endpoint> **TO** <signal route endpoint> [**WITH** <signal list>] <end>

<signal route endpoint> ::=

{ <process identifiant> | <service identifiant> | **ENV** } [**VIA** <gate>]

Information complémentaire:

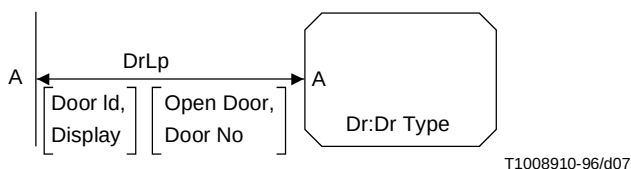
En SDL96 seulement, si le mot clé **InvisibleName** est présent, alors le nom de l'acheminement de signal ne doit pas apparaître dans le GR. Le nom est seulement présent dans le PR afin de pouvoir référencer cet acheminement dans une instruction de connexion **CONNECT**. Le mot clé **InvisibleName** ne doit pas apparaître dans le CIF généré à partir d'une description SDL92.

Le premier élément dans la liste des points est situé sur le rectangle englobant le symbole, il correspond à l'instruction **FROM** <signal route endpoint> du premier trajet <signal route path>. Le dernier élément dans la liste des points est situé sur le rectangle englobant le symbole, il correspond à l'instruction **TO** <signal route endpoint> du premier trajet <signal route path>.

Les flèches de l'acheminement sont placées automatiquement lors de l'analyse du CIF.

Le positionnement du texte pour l'accès associé à l'instruction **VIA** est spécifié soit par la règle <gate: A23>, soit par la règle <gate reference: B19>.

Exemple:




```

/* CIF SignalRoute (500,400),(300,400) */
/* CIF TextPosition (390,410) */
/* CIF TextPosition (490,410) SignalList1 */
/* CIF TextPosition (290,410) SignalList2 */
SIGNALROUTE DrLp
FROM Dr VIA A TO env VIA A
WITH DoorId,
Display;
FROM env VIA A TO Dr VIA A
WITH OpenDoor,
DoorNo;

```

A23 Accès <gate: A23>:

```

/* CIF Gate <pointlist: B22> [ Dashed ] */
[ <gate name text position: B25> ]
[ <first signallist text position: B6> ]
[ <second signallist text position: B7> ]
[ <gate constraint symbol: B5> ]
<gate definition>

```

Cette règle est utilisée par <CIF descriptor: A18>.

<gate definition> ::=

GATE <gate name> [**ADDING**] <gate constraint> <end> [<gate constraint> <end>]

<gate constraint> ::=

{ **OUT** [**TO** <textual endpoint constraint>] | **IN** [**FROM** <textual endpoint constraint>] } [**WITH** <signal list>]

<textual endpoint constraint> ::=

[**ATLEAST**] <identifieur>

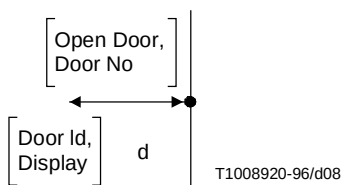
Information complémentaire:

La liste <pointlist: B22> doit contenir deux points. Le premier point est situé sur le cadre du diagramme. Le second point correspond à l'emplacement où l'accès est défini. Si une contrainte <textual endpoint constraint> est présente, le second point sera situé sur le rectangle englobant le symbole correspondant à cette contrainte.

Le mot clé **Dashed** doit être spécifié, si le mot clé **ADDING** est utilisé dans le SDL-PR.

La contrainte <gate constraint symbol: B5> doit être spécifiée, si la contrainte <textual endpoint constraint> est présente.

Exemple 1:

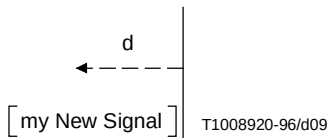


```

/* CIF Gate (500,400),(300,400) */
/* CIF TextPosition (390,410) */
/* CIF TextPosition (490,410) SignalList1 */
/* CIF TextPosition (290,410) SignalList2 */
GATE d OUT
WITH DoorId,
Display;
IN
WITH OpenDoor,
DoorNo;

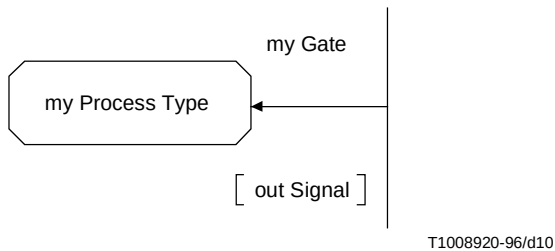
```

Exemple 2:



```
/* CIF Gate (500,400), (300,400) Dashed */
GATE d ADDING OUT
WITH myNewSignal;
```

Exemple 3:



```
/* CIF Gate (500,400), (300,400) */
/* CIF Process (100,350), (200,100) */
GATE myGate OUT TO myProcessType
WITH outSignal;
```

A24 Connexion <connect: A24>:

```
/* CIF Connect */
[ <text position: B25> ]
{ <channel to route connection> | <signal route to route connection> | <channel connection> }
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<channel to route connection> ::=
CONNECT <channel identifiers> AND <signal route identifiers> <end>

<channel identifiers> ::=
<channel identifier> { , <channel identifier> }*

<signal route identifiers> ::=
<signal route identifier> { , <signal route identifier> }*

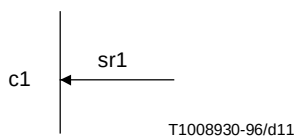
<signal route to route connection> ::=
CONNECT <external signal route identifiers> AND <signal route identifiers> <end>

<channel connection> ::=
CONNECT <channel identifiers> AND <subchannel identifiers> <end>
```

Information complémentaire:

La position <text position: B25> spécifie la position du texte des identificateurs <channel identifiers> ou <external signal route identifiers>, c'est-à-dire là où le texte doit être mis à l'extérieur du cadre.

Exemple:



```
/* CIF Connect */
/* CIF TextPosition (800,50) */
CONNECT c1 AND sr1;
```

A25 Extension de texte <text extension: A25>:

```
<text outside extension>
/* CIF TextExtension <position and size: B24>
[ { Left | Right } ] */
[ <text position: B25> ]
[ <line: B20> ]
<text inside extension>
/* CIF End TextExtension */
```

Cette règle n'est utilisée par aucune autre règle: voir "Information complémentaire".

Information complémentaire:

Le mot clé **Left** signifie que l'ouverture du symbole est côté gauche. Le mot clé **Right** signifie que l'ouverture du symbole est côté droit. L'option par défaut est **Right**.

Si le mot clé **Left** est présent, les positions du symbole et du texte sont calculées à partir du coin supérieur *droit*.

La ligne <line: B20> correspond à la ligne reliant le symbole d'extension de texte à son symbole SDL. S'il n'y a pas de ligne spécifiée, alors elle sera tracée automatiquement. Si elle est spécifiée, le premier point de la liste des points correspond au point de départ de la ligne à partir du rectangle englobant le symbole d'extension de texte. Le dernier point de la liste correspond au point d'arrivée de la ligne sur le rectangle englobant le symbole SDL portant cette extension de texte.

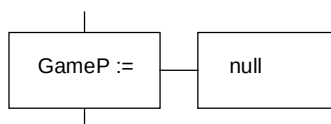
Un caractère de saut de ligne placé au début ou à la fin d'une directive CIF d'extension de texte ne sera pas considéré comme faisant partie du texte de cette extension.

Les extensions <text extension: A25> et commentaires <comment: A26> doivent être placés avant la fin <end> de la manière suivante: extensions et commentaires peuvent être associés à toutes les règles de <block symbol: A30> à <select symbol: A69>.

Exemple 1 (exemple informel avec un symbole de tâche):

```
/* CIF Task (800,550) */
TASK 'first part of task text that will be in the task symbol'
/* CIF TextExtension (1100,550) */
'last part of task text that will be in the TextExtension symbol';
/* CIF End TextExtension */
;
```

Exemple 2:



T1008940-96/d12

```
/* CIF Task (800,550) */
TASK GameP :=
/* CIF TextExtension (1100,550) */
/* CIF Line (1100,600),(1000,600) */
null
/* CIF End TextExtension */
;
```

A26 Commentaire <comment: A26>:

```
/* CIF Comment <position and size: B24> [ Left | Right ] [ Dashed ] */
[ <text position: B25> ]
[ <dashed line: B21> ]
<comment> <end>
```

Cette règle n'est utilisée par aucune autre règle: voir "Information complémentaire".

```
<comment> ::=
COMMENT <character string>
```

Information complémentaire:

Le mot clé **Left** signifie que l'ouverture du symbole est côté gauche. Le mot clé **Right** signifie que l'ouverture du symbole est côté droit. L'option par défaut est **Right**.

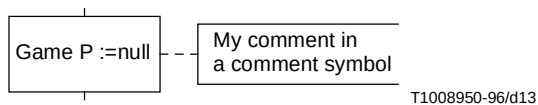
Si le mot clé **Left** est présent, la position du symbole (et du texte) est calculée à partir du coin supérieur *droit*.

Si le mot clé **Dashed** est présent, alors le symbole de commentaire doit être tracé en pointillé (comme un symbole <comment symbol2> en SDL96). Dans le cas contraire, le symbole de commentaire doit être tracé en trait plein (comme un symbole <comment symbol> en SDL92).

La ligne <dashed line: B21> correspond à la ligne reliant le symbole de commentaire à son symbole SDL. S'il n'y a pas de ligne spécifiée, alors elle sera tracée automatiquement. Si elle est spécifiée, le premier point de la liste des points correspond au point de départ de la ligne à partir du rectangle englobant le symbole de commentaire. Le dernier point de la liste correspond au point d'arrivée de la ligne sur le rectangle englobant le symbole SDL portant ce commentaire.

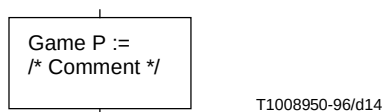
La règle <text extension: A25> explique la façon dont cette création de format CIF est utilisée.

Exemple 1:



```
/* CIF Task (800,550) */
task GameP:=null
/* CIF Comment (1100,550) */
/* CIF Line (1100,600),(1000,600) Dashed */
COMMENT 'My comment in
a comment symbol';
```

Autre présentation:



```
* CIF Task (800,550) */
task GameP:=null
/* Comment */;
```

Exemple 2:

```
/* CIF Task (800,550) */
task GameP :=
/* CIF TextExtension (1100,550) */
/* CIF Line (1100,600),(1000,600) */
null
/* CIF End TextExtension */
/* CIF Comment (1100,750) */
/* CIF Line (1100,800),(1000,600) Dashed */
COMMENT 'My comment in a comment symbol';
```

A27 Ligne de création <create line: A27>:

```
/* CIF CreateLine <pointlist: B22> */
```

Cette règle est utilisée par <CIF descriptor: A18>.

Information complémentaire:

Le premier point de la liste est situé sur le rectangle symbolisant le processus créateur. Le dernier point de la liste est situé sur le rectangle symbolisant le processus créé.

Ce commentaire CIF peut être placé avant ou après tout symbole ou ligne de CIF ou de PR à l'intérieur du diagramme, comme si c'était une entité <entity in block>.

Exemple:

```

/* CIF DefaultSize (200,100) */
/* CIF Process (200,500) */
PROCESS Main(1,1) REFERENCED;
/* CIF Process (500,500) */
PROCESS Game(0,1) REFERENCED;
/* CIF CreateLine (400,550),(500,550) */

```

A28 Ligne de flux <flow line: A28>:

[<line: B20>]

Cette règle est utilisée par <CIF descriptor: A18>.

Information complémentaire:

Ce commentaire CIF est facultatif. S'il n'est pas présent pour décrire la ligne de flux, alors cette ligne est tracée automatiquement.

Le premier point de la liste est situé sur le rectangle englobant le symbole à partir duquel la ligne de flux débute. Le dernier point de la liste est situé sur le rectangle englobant le symbole sur lequel la ligne de flux arrive.

Ce commentaire peut être placé avant ou après tout symbole ou ligne de CIF ou de PR à l'intérieur du corps <process body> du diagramme.

Une ligne de flux connectée à une autre ligne de flux doit contenir la liste complète des points, du point situé sur le rectangle englobant le symbole de départ, au point situé sur le rectangle englobant le symbole d'arrivée.

Les flèches sur les lignes de flux sont implicites. Les lignes de flux partant de symboles de décision ou d'option de transition doivent utiliser la règle de production de ligne <answer flow line: A29>, règle CIF qui est obligatoire.

Exemple:

```

/* CIF Start (300,100) */
START;
/* CIF Line (400,200),(400,250) */
/* CIF Set (300,250) */
Set(Now+1, T);

```

A29 Réponse de décision <answer flow line: A29>:

```

/* CIF Answer [ { Right | Left } ] [ InvisibleBrackets ] */
[ <line: B20> ] [ <text position: B25> ]
{ <answer part> | <else part> }

```

Cette règle est utilisée par <CIF descriptor: A18>.

```

<answer part> ::=
( [ <answer> ] ) : [ <transition> ]

```

```

<answer> ::=
<range condition> | <informal text>

```

```

<else part> ::=
ELSE : [ <transition> ]

```

Information complémentaire:

Ce commentaire CIF doit être utilisé pour décrire les lignes de flux positionnées après des décisions ou des options de transition.

Si la règle <line: B20> est présente:

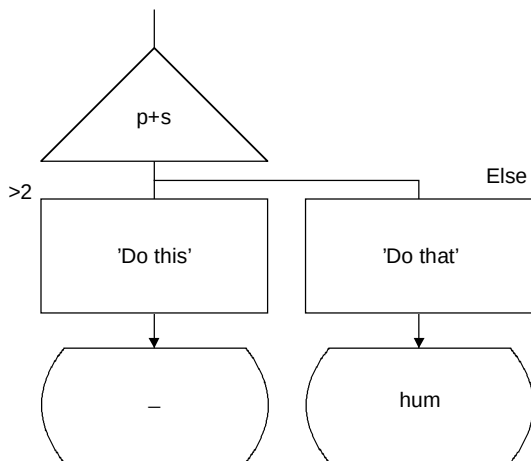
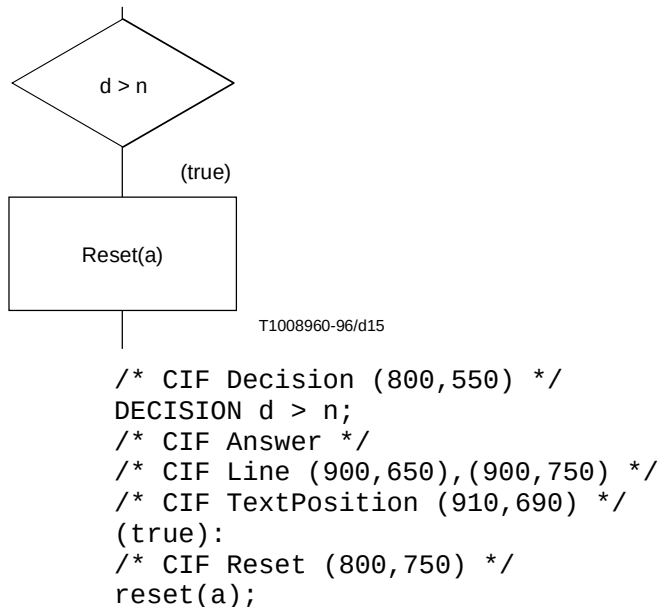
les mots clés **Right** et **Left** n'ont pas de sens. La liste de points de la ligne de flux indique d'où débute la ligne à partir du symbole de décision. Les mêmes règles que pour la ligne de flux <flow line: A28> sont applicables.

Si la règle <line: B20> n'est pas présente:

le mot clé **Right** précise que la ligne de flux démarre du côté droit du symbole de décision (ou du coin inférieur droit du symbole d'option de transition). Le mot clé **Left** précise que la ligne de flux démarre du côté gauche du symbole de décision (ou du coin inférieur gauche du symbole d'option de transition). Par défaut, la ligne de flux démarre exactement en dessous du symbole de décision (ou du symbole d'option de transition). Le reste de la ligne de flux est tracé automatiquement.

Si le mot clé **InvisibleBrackets** est présent, les caractères "(" et ")" qui peuvent être trouvés dans le PR ne sont pas visibles graphiquement.

Exemples:



T1008960-96/d16

```

/* CIF Alternative (800,550) */
ALTERNATIVE p + s;
/* CIF Answer InvisibleBrackets */
(>2):
/* CIF Task (800,750) */
TASK 'Do this';
/* CIF NextState (800,950) */
NEXTSTATE -;
/* CIF Answer */
ELSE:
/* CIF Task (1100,750) */
TASK 'Do that';
/* CIF NextState (1100,950) */
NEXTSTATE hum;

```

A30 Symbole de bloc <block symbol: A30>:

```
<block symbol rectangle: B15>  
[ <text position: B25> ]  
{ <textual block reference> |  
  { <gate reference: B19>* <textual typebased system definition> } }
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<textual block reference> ::=  
BLOCK <name> REFERENCED <end>  
  
<textual typebased block definition> ::=  
BLOCK <typebased block heading> <end>  
  
<typebased block heading> ::=  
<block name> [ <number of block instances> ] : <block type expression>  
  
<number of block instances> ::=  
( <natural simple expression> )
```

Information complémentaire:

Les références <gate reference: B19> sont facultatives et ne sont utilisées que pour préciser la position du texte pour les références d'accès associées au symbole de bloc. Si une référence <gate reference: B19> est omise, la position du texte pour cette référence d'accès sera calculée automatiquement. En PR, le nom d'une référence d'accès est indiqué en relation avec les parties de PR contenant les déclarations de canaux et d'acheminements de signaux.

Exemple 1:

```
/* CIF Block (800,550) */  
/* CIF TextPosition (810,560) */  
BLOCK myBlock REFERENCED;
```

Exemple 2:

```
/* CIF Block (800,550) */  
/* CIF GateReference (900,550) */  
/* CIF TextPosition (890,500) */  
BLOCK myBlocks(2):myBlockType;
```

A31 Symbole de bloc en pointillé <dashed block symbol: A31>:

```
/* CIF Block <position and size: B24> Dashed <block name> */  
[ <text position: B25> ]  
<gate reference: B19>*
```

Cette règle est utilisée par <CIF descriptor: A18>.

Information complémentaire:

Ce commentaire CIF peut être placé partout où une référence <textual block reference> est permise. Il doit y avoir un symbole <dashed block symbol: A31> par définition <existing typebased block definition> dans le SDL-GR. La règle <gate reference: B19> est expliquée dans la règle <block symbol: A30>.

Exemple:

```
/* CIF Block (800,550) Dashed myBlock */
```

A32 Symbole de processus <process symbol: A32>:

```
<process symbol rectangle: B16>  
[ <text position: B25> ]  
{ <textual process reference> |  
  { <gate reference: B19>* <textual typebased process definition> } }
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<textual process reference> ::=  
PROCESS <name> [ <number of process instances> ] REFERENCED <end>
```

<textual typebased process definition> ::=
PROCESS <typebased process heading> <end>

<typebased process heading> ::=
<process name> [<number of process instances>] : <process type expression>

Information complémentaire:

La règle <gate reference: B19> est expliquée dans la règle <block symbol: A30>.

Exemple 1:

```
/* CIF Process (800,550) */  
PROCESS myProcess REFERENCED;
```

Exemple 2:

```
/* CIF Process (800,550) */  
PROCESS myProcess (1,1):myProcessType;
```

A33 Symbole de processus en pointillé <dashed process symbol: A33>:

```
/* CIF Process <position and size: B24> Dashed <process name> */  
[ <text position: B25> ]  
<gate reference: B19>*
```

Cette règle est utilisée par <CIF descriptor: A18>.

Information complémentaire:

Ce commentaire CIF peut être placé partout où une référence <textual process reference> est permise. Il doit y avoir un symbole <dashed process symbol: A33> par définition <existing typebased process definition> dans le SDL-GR. La règle <gate reference: B19> est expliquée dans la règle <block symbol: A30>.

Exemple:

```
/* CIF Process (800,550) Dashed myProcess */
```

A34 Symbole de service <service symbol: A34>:

```
<service symbol rectangle: B17>  
[ <text position: B25> ]  
{ <textual service reference> |  
  { <gate reference: B19>* <textual typebased service definition> } }
```

Cette règle est utilisée par <CIF descriptor: A18>.

<textual service reference> ::=
SERVICE <name> **REFERENCED** <end>

<textual typebased service definition> ::=
SERVICE <typebased service heading> <end>

<typebased service heading> ::=
<service name> : <service type expression>

Information complémentaire:

La règle <gate reference: B19> est expliquée dans la règle <block symbol: A30>.

Exemple 1:

```
/* CIF Service (800,550) */  
SERVICE myService REFERENCED;
```

Exemple 2:

```
/* CIF Service (800,550) */  
SERVICE myService:myServiceType;
```


A35 Symbole de service en pointillé <dashed service symbol: A35>:

```
/* CIF Service <position and size: B24> Dashed <service name> */  
[ <text position: B25> ]  
<gate reference: B19>*
```

Cette règle est utilisée par <CIF descriptor: A18>.

Information complémentaire:

Ce commentaire CIF peut être placé partout où une référence <textual service reference> est permise. Il doit y avoir un symbole <dashed service symbol: A35> par définition <existing typebased service definition> dans le SDL-GR. La règle <gate reference: B19> est expliquée dans la règle <block symbol: A30>.

Exemple:

```
/* CIF Service (800,550) Dashed myService */
```

A36 Symbole de type de système <system type symbol: A36>:

```
/* CIF SystemType <position and size: B24> */  
[ <text position: B25> ]  
<textual system type reference>
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<textual system type reference> ::=  
SYSTEM TYPE <name> REFERENCED <end>
```

Exemple:

```
/* CIF SystemType (800,550) */  
SYSTEM TYPE mySystemType REFERENCED;
```

A37 Symbole de type de bloc <block type symbol: A37>:

```
/* CIF BlockType <position and size: B24> */  
[ <text position: B25> ]  
<textual block type reference>
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<textual block type reference> ::=  
[ <virtuality> ] BLOCK TYPE <name> REFERENCED <end>
```

Exemple:

```
/* CIF BlockType (800,550) */  
VIRTUAL BLOCK TYPE myBlockType REFERENCED;
```

A38 Symbole de type de processus <process type symbol: A38>:

```
/* CIF ProcessType <position and size: B24> */  
[ <text position: B25> ]  
<textual process type reference>
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<textual process type reference> ::=  
[ <virtuality> ] PROCESS TYPE <name> REFERENCED <end>
```

Exemple:

```
/* CIF ProcessType (800,550) */  
PROCESS TYPE myProcessType REFERENCED;
```

A39 Symbole de type de service <service type symbol: A39>:

```
/* CIF ServiceType <position and size: B24> */  
[ <text position: B25> ]  
<textual service type reference>
```

Cette règle est utilisée par <CIF descriptor: A18>.

<textual service type reference> ::=
[<virtuality>] **SERVICE TYPE** <name> **REFERENCED** <end>

Exemple:

```
/* CIF ServiceType (800,550) */  
SERVICE TYPE myServiceType REFERENCED;
```

A40 Symbole de sous-structure de bloc <block substructure symbol: A40>:

```
/* CIF BlockSubstructure <position and size: B24> */  
[ <text position: B25> ]  
<textual block substructure reference>
```

Cette règle est utilisée par <CIF descriptor: A18>.

<textual block substructure reference> ::=
SUBSTRUCTURE <name> **REFERENCED** <end>

Exemple:

```
/* CIF BlockSubstructure (800,550) */  
SUBSTRUCTURE mySubstructure REFERENCED;
```

A41 Symbole de procédure <procedure symbol: A41>:

```
/* CIF Procedure <position and size: B24> */  
[ <text position: B25> ]  
<textual procedure reference>
```

Cette règle est utilisée par <CIF descriptor: A18>.

<textual procedure reference> ::=
<procedure preamble> **PROCEDURE** <name> **REFERENCED** <end>

<procedure preamble> ::=
[<virtuality>] [**EXPORTED** [**AS** <remote procedure identifier>]]

Exemple:

```
/* CIF Procedure (800,550) */  
VIRTUAL PROCEDURE myProcedure REFERENCED;
```

A42 Symbole d'opérateur <operator symbol: A42>:

```
/* CIF Operator <name> <position and size: B24> */  
[ <text position: B25> ]
```

Cette règle est utilisée par <CIF descriptor: A18>.

Information complémentaire:

Ce commentaire peut être placé partout où une entité <entity in package>, <entity in system>, <entity in block>, <entity in process>, <entity in service> ou <entity in procedure> est permise. Le symbole d'opérateur n'a pas de relation graphique directe avec la référence <textual operator reference> (qui est présente en tant que texte dans un symbole de texte).

Notons que cette règle est un compromis car le symbole d'opérateur n'existe pas en SDL-GR. Pour être complète, l'identification d'un opérateur ne se limite pas à son nom. Il faut aussi prendre en compte ses qualificatifs, ses paramètres et son résultat. Une difficulté supplémentaire est que le nom de l'opérateur peut être remplacé par un opérateur <quoted operator>.

Exemple:

```
/* CIF Operator myOperator (800,550) */
```

A43 Symbole de départ <start symbol: A43>:

```
/* CIF Start <position and size: B24> */  
[ <text position: B25> ]  
<beginning of start>
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<beginning of start> ::=  
START [ <virtuality> ] <end>
```

Information complémentaire:

Ce commentaire CIF doit être utilisé dans des diagrammes de processus ou de types de processus. Les diagrammes de procédures et d'opérateurs doivent utiliser la règle <procedure start symbol: A54>.

Exemple:

```
/* CIF Start (800,550) */  
START;
```

A44 Symbole d'arrêt <stop symbol: A44>:

```
/* CIF Stop <position and size: B24> */  
<stop> <end>
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<stop> ::=  
STOP
```

Exemple:

```
/* CIF Stop (800,550) */  
STOP;
```

A45 Symbole d'état <state symbol: A45>:

```
/* CIF State <position and size: B24> */  
[ <text position: B25> ]  
<beginning of state>
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<beginning of state> ::=  
STATE <state list> <end>
```

```
<state list> ::=  
{ <state name> { , <state name> } * } | <asterisk state list>
```

Exemple: des exemples d'utilisation de ce commentaire CIF sont présentés avec la règle <nextstate symbol: A46>.

A46 Symbole d'état suivant <nextstate symbol: A46>:

```
/* CIF NextState <position and size: B24> */  
[ <text position: B25> ]  
<nextstate> <end>
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<nextstate> ::=  
NEXTSTATE <nextstate body>
```

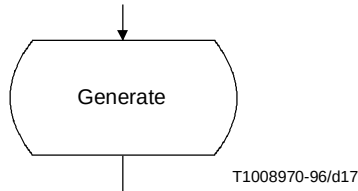
```
<nextstate body> ::=  
{ <state name> | <dash nextstate> }
```

```
<dash nextstate> ::=  
<hyphen>
```

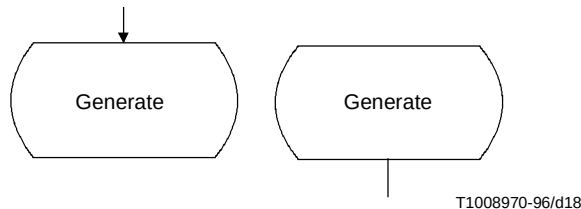
Information complémentaire:

Il y a un commentaire CIF par état et par état suivant présent dans le PR. Cela signifie qu'il y aura deux commentaires CIF pour un symbole GR représentant à la fois l'état et l'état suivant vus dans le PR. Un outil lisant du CIF devra déterminer si un état et un état suivant correspondent au même symbole GR, en comparant les coordonnées des deux symboles données dans les commentaires CIF.

Exemples:



```
/* CIF NextState (800,550) */  
NEXTSTATE Generate;  
/* CIF State (800,550) */  
STATE Generate;
```



```
/* CIF NextState (800,550) */  
NEXTSTATE Generate;  
/* CIF State (1100,550) */  
STATE Generate;
```

A47 Symbole de sauvegarde <save symbol: A47>:

```
/* CIF Save <position and size: B24> */  
[ <text position: B25> ]  
<basic save part>
```

Cette règle est utilisée par <CIF descriptor: A18>.

<basic save part> (en SDL92) ::=

(identique à la partie <save part> en SDL96. Les parties <basic save part> en SDL92 et <save part> en SDL96 doivent utiliser ce commentaire CIF)

SAVE [<virtuality>] <save list> <end>

<save list> ::=

{ <signal list> | <asterisk save list> }

Exemple:

```
/* CIF Save (800,550) */  
SAVE mySignal;
```

A48 Symbole de tâche <task symbol: A48>:

```
/* CIF Task <position and size: B24> */  
[ <text position: B25> ]  
<task> <end>
```

Cette règle est utilisée par <CIF descriptor: A18>.

```

<task> ::=
TASK <task body>

<task body> ::=
{ <assignment statement> { , <assignment statement> }* } |
{ <informal text> { , <informal text> }* }

<informal text> ::=
<character string>

<character string> ::=
<apostrophe> { <alphanumeric> | <other character> | <special> | <full stop> | <underline> | <space> |
{ <apostrophe><apostrophe> } }* <apostrophe>

```

Information complémentaire:

Il y a trois cas où un symbole de tâche peut être décrit par un autre commentaire CIF que celui-ci: les symboles de tâches GR contenant des armements <set> doivent utiliser la règle <set symbol: A49>; les symboles de tâches GR contenant des réinitialisations <reset> doivent utiliser la règle <reset symbol: A50>; les symboles de tâches GR contenant des exportations <export> doivent utiliser la règle <export symbol: A51>.

Exemple:

```

/* CIF Task (800,550) */
TASK myVariable := 0;

```

A49 Symbole d'armement <set symbol: A49>:

```

/* CIF Set <position and size: B24> */
[ <text position: B25> ]
<set> <end>

```

Cette règle est utilisée par <CIF descriptor: A18>.

```

<set> ::=
SET <set statement> { , <set statement> }*

<set statement> ::=
( [ <time expression> , ] <timer identifier> [ ( <expression list> ) ] )

<expression list> ::=
<expression> { , <expression> }*

```

Information complémentaire:

Un symbole d'armement est un symbole de tâche GR contenant la règle <set>.

Exemple:

```

/* CIF Set (800,550) */
SET (Now+1, myTime);

```

A50 Symbole de réinitialisation <reset symbol: A50>:

```

/* CIF Reset <position and size: B24> */
[ <text position: B25> ]
<reset> <end>

```

Cette règle est utilisée par <CIF descriptor: A18>.

```

<reset> ::=
RESET <reset statement> { , <reset statement> }*

<reset statement> ::=
<timer identifier> [ <expression list> ]

```

Information complémentaire:

Un symbole de réinitialisation est un symbole de tâche GR contenant la règle <reset>.

Exemple:

```
/* CIF Reset (800,550) */
RESET T;
```

A51 Symbole d'exportation <export symbol: A51>:

```
/* CIF Export <position and size: B24> */
[ <text position: B25> ]
<export> <end>
```

Cette règle est utilisée par <CIF descriptor: A18>.

<export> ::=

EXPORT (<variable identifieur> { , <variable identifieur> }*)

Information complémentaire:

Un symbole d'exportation est un symbole de tâche GR contenant la règle <export>.

Exemple:

```
/* CIF Export (800,550) */
Export (myVariable1, myVariable2);
```

A52 Symbole de demande de création <create symbol: A52>:

```
/* CIF Create <position and size: B24> */
[ <text position: B25> ]
<create request> <end>
```

Cette règle est utilisée par <CIF descriptor: A18>.

<create request> ::=

CREATE <create body>

<create body> ::=

{ <process identifieur> | **THIS** } [<actual parameters>]

<actual parameters> ::=

([<expression>] { , [<expression>] }*)

Exemple:

```
/* CIF Create (800,550) */
CREATE Game;
```

A53 Symbole d'appel de procédure <procedure call symbol: A53>:

```
/* CIF ProcedureCall <position and size: B24> */
[ <text position: B25> ]
<procedure call> <end>
```

Cette règle est utilisée par <CIF descriptor: A18>.

<procedure call> ::=

CALL <procedure call body>

<procedure call body> ::=

[**THIS**] <procedure identifieur> [<actual parameters>]

Exemple:

```
/* CIF ProcedureCall (800,550) */
CALL myProcedure;
```

A54 Symbole de début de procédure <procedure start symbol: A54>:

```
/* CIF ProcedureStart <position and size: B24> */  
[ <text position: B25> ]  
<beginning of start>
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<beginning of start> ::=  
START [ <virtuality> ] <end>
```

Information complémentaire:

Ce commentaire CIF doit être utilisé pour les diagrammes de procédures et d'opérateurs. Les diagrammes de processus et de types de processus doivent utiliser la règle <start symbol: A43>.

Exemple:

```
/* CIF ProcedureStart (800,550) */  
START;
```

A55 Symbole de retour <return symbol: A55>:

```
/* CIF Return <position and size: B24> */  
[ <text position: B25> ]  
<return> <end>
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<return> ::=  
RETURN [ <expression> ]
```

Exemple:

```
/* CIF Return (800,550) */  
RETURN myReturnValue;
```

A56 Symbole de décision <decision symbol: A56>:

```
/* CIF Decision <position and size: B24> */  
[ <text position: B25> ]  
<beginning of decision>
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<beginning of decision> ::=  
DECISION <question> <end>
```

```
<question> ::= <question expression> | <informal text> | ANY
```

Information complémentaire:

Une ligne de flux partant d'un symbole de décision doit être décrite par la règle <answer flow line: A29>.

Exemple (sans ligne de flux):

```
/* CIF Decision (800,550) */  
DECISION DoorIndex > NoOfDoors;
```

A57 Symbole de signal continu <continuous signal symbol: A57>:

```
/* CIF ContinuousSignal <position and size: B24> */  
[ <text position: B25> ]  
<beginning of continuous signal>
```

Cette règle est utilisée par <CIF descriptor: A18>.

<beginning of continuous signal> ::=

PROVIDED [<virtuality>] <boolean expression> <end> [**PRIORITY** <integer literal name> <end>]

Exemple:

```
/* CIF ContinuousSignal (800,550) */  
PROVIDED level > 5;
```

A58 Symbole de condition de validation <enabling condition symbol: A58>:

```
/* CIF EnablingCondition <position and size: B24> */  
[ <text position: B25> ]  
<enabling condition>
```

Cette règle est utilisée par <CIF descriptor: A18>.

<enabling condition> ::=

PROVIDED <boolean expression> <end>

Exemple:

```
/* CIF EnablingCondition (800,550) */  
PROVIDED level > 5;
```

A59 Symbole d'option de transition <transition option symbol: A59>:

```
/* CIF TransitionOption <position and size: B24> */  
[ <text position: B25> ]  
<beginning of transition option>
```

Cette règle est utilisée par <CIF descriptor: A18>.

<beginning of transition option> ::=

ALTERNATIVE <alternative question> <end>

<alternative question> ::=

<simple expression> | <informal text>

Information complémentaire:

Une ligne de flux partant d'un symbole d'option de transition doit être décrite par la règle <answer flow line: A29>.

Exemple:

```
/* CIF TransitionOption (800,550) */  
ALTERNATIVE level;
```

A60 Symbole de branchement <join symbol: A60>:

```
/* CIF Join { <position and size: B24> | Invisible } */  
[ <text position: B25> ]  
<join> <end>
```

Cette règle est utilisée par <CIF descriptor: A18>.

<join> ::=

JOIN <connector name>

Information complémentaire:

Le mot clé **Invisible** signifie que le branchement PR ne doit pas être tracé comme un symbole dans le GR. Ce type de branchement en PR sert à spécifier qu'une ligne de flux se termine sur un symbole déjà décrit (c'est-à-dire sur le symbole suivant l'étiquette pointée par le branchement). Voir aussi <label symbol: A64>.

Exemple 1:

```
/* CIF Join Invisible */  
JOIN myInvisibleLabel;
```


Exemple 2:

```
/* CIF Join (800,550) */
JOIN myLabel;
```

A61 Symbole d'appel de macro <macro call symbol: A61>:

```
/* CIF MacroCall <position and size: B24> */
[ <text position: B25> ]
[ <inlet text: B10> ]
[ <outlet text: B11> ]
<macro call>
```

Cette règle est utilisée par <CIF descriptor: A18>.

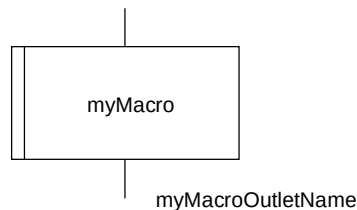
```
<macro call> ::=
MACRO <macro name> [ <macro call body> ] <end>

<macro call body> ::=
( <macro actual parameter> { , <macro actual parameter> }* )

<macro actual parameter> ::=
{ <lexical unit> }*
```

Information complémentaire:

Le texte <inlet text: B10> doit être présent si le symbole d'appel de macro est associé à un label d'accès entrant de macro. Le texte <outlet text: B11> doit être présent si le symbole d'appel de macro est associé à un label d'accès sortant de macro. Pour plus de détails, se reporter à la règle <macro diagram start: A16>.

Exemple:

T1008980-96/d19

```
/* CIF MacroCall (800,550) */
/* CIF OutletText myMacroOutletName */
MACRO myMacro;
```

A62 Symbole d'accès entrant de macro <macro inlet symbol: A62>:

```
/* CIF MacroInlet <position and size: B24> */
[ <inlet text: B10> ]
<macro body>
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<macro body> ::=
{ <lexical unit> | <formal name> }*
```

Information complémentaire:

Le texte <inlet text: B10> doit être présent si le symbole d'accès entrant de macro est associé à un label de macro. Pour plus de détails, se reporter à la règle <macro diagram start: A16>.

Exemple (accès entrant sans étiquette, suivi d'un symbole de sortie):

```
/* CIF MacroInlet (800,550) */
/* CIF Output (800,750) */
OUTPUT outSignal;
```

A63 Symbole d'accès sortant de macro <macro outlet symbol: A63>:

```
/* CIF MacroOutlet <position and size: B24> */  
[ <outlet text: B11> ]  
<diagram end: A17>  
<end of macro definition>
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<end of macro definition> ::=  
ENDMACRO [ <macro name> ] <end>
```

Information complémentaire:

Le texte <outlet text: B11> doit être présent si le symbole d'accès sortant de macro est associé à un label de macro. Pour plus de détails, se reporter à la règle <macro diagram start: A16>.

Exemple:

```
/* CIF MacroOutlet (800,950) */  
/* CIF End MacroDiagram */  
ENDMACRO myMacro;
```

A64 Symbole d'étiquette <label symbol: A64>:

```
/* CIF Label { <position and size: B24> | Invisible } */  
[ <text position: B25> ]  
( <label> | <beginning of free action> )
```

Cette règle est utilisée par <CIF descriptor: A18>.

```
<label> ::=  
<connector name>;
```

```
<beginning of free action> ::=  
CONNECTION <beginning of transition>
```

Information complémentaire:

Le mot clé **Invisible** signifie que l'étiquette PR ne doit pas être tracée comme un symbole dans le GR. Ce type d'étiquette en PR sert à spécifier qu'une ligne de flux se termine sur un symbole déjà décrit (c'est-à-dire sur le symbole suivant l'étiquette). Voir aussi <join symbol: A60>.

Le commentaire CIF correspondant à la première étiquette <label> dans un début d'action <beginning of free action> est placé avant le début d'action <beginning of free action>, se reporter à l'exemple 3.

Exemple 1:

```
/* CIF Label Invisible */  
myInvisibleLabel:
```

Exemple 2:

```
/* CIF Label (800,550),(100,100) */  
myVisibleLabel:
```

Exemple 3:

```
/* CIF Label (800,550), (100,100) */  
CONNECTION myLabel:
```

A65 Symbole d'entrée <input symbol: A65>:

```
/* CIF Input <position and size: B24> [ { Left | Right } ] */  
[ <text position: B25> ]  
{ <beginning of basic input part> | <beginning of spontaneous transition> | <beginning of remote procedure input transition> }
```

Cette règle est utilisée par <CIF descriptor: A18>.

<beginning of basic input part> (en SDL92) ::=

(... identique à la partie <beginning of input part> de SDL96. Les parties <beginning of basic input part> de SDL92 et <beginning of input part> de SDL96 doivent, toutes les deux, utiliser ce commentaire CIF)

INPUT [<virtuality>] <input list> <end>

<input list> ::=

<asterisk> | { <stimulus> { , <stimulus> }* }

<stimulus> (en SDL92) ::=

{ <signal identifier> | <timer identifier> } [([<variable>] { , [<variable>] }*)]

<stimulus> (en SDL96) ::=

<signal list item> [([<variable>] { , [<variable>] }*)]

<beginning of spontaneous transition> ::=

INPUT [<virtuality>] **NONE** <end>

<beginning of remote procedure input transition> (en SDL92) ::=

INPUT [<virtuality>] **PROCEDURE** <remote procedure identifier list> <end>

<remote procedure identifier list> ::=

<remote procedure identifier> { , <remote procedure identifier> }*

Information complémentaire:

Le mot clé **Left** signifie que la flèche sera tracée côté gauche. Le mot clé **Right** signifie que la flèche sera tracée côté droit. L'option par défaut est **Right**.

Exemple:

```
/* CIF Input (800,550) Left */  
INPUT mySignal;
```

A66 Symbole d'entrée prioritaire <priority input symbol: A66>:

/* **CIF PriorityInput** <position and size: B24>

[{ **Left** | **Right** }] */

[<text position: B25>]

<beginning of priority input>

Cette règle est utilisée par <CIF descriptor: A18>.

<beginning of priority input> ::=

PRIORITY INPUT [<virtuality>] <priority input list> <end>

<priority input list> ::=

<stimulus> { , <stimulus> }*

Information complémentaire:

Le mot clé **Left** signifie que la flèche sera tracée côté gauche. Le mot clé **Right** signifie que la flèche sera tracée côté droit. L'option par défaut est **Right**.

Exemple:

```
/* CIF PriorityInput (800,550) Left */  
PRIORITY INPUT mySignal;
```

A67 Symbole de sortie <output symbol: A67>:

/* **CIF Output** <position and size: B24> [{ **Left** | **Right** }] */

[<text position: B25>]

<output> <end>

Cette règle est utilisée par <CIF descriptor: A18>.

```

<output> ::=
OUTPUT <output body>

<output body> ::=
<signal identifier> [ <actual parameters> ] { , <signal identifier> [ <actual parameters> ] }* [ TO <destination> ]
[ VIA [ ALL ] <via path> ]

<destination> ::=
<Pid expression> | <process identifier> | THIS

<via path> ::=
<via path element> { , <via path element> }*

<via path element> ::=
<signal route identifier> | <channel identifier> | <gate identifier>

```

Information complémentaire:

Le mot clé **Left** signifie que la flèche sera tracée côté gauche. Le mot clé **Right** signifie que la flèche sera tracée côté droit. L'option par défaut est **Right**.

Exemple:

```

/* CIF Output (800,550) */
OUTPUT mySignal;

```

A68 Symbole de texte <text symbol: A68>:

```

/* CIF Text <position and size: B24> */
[ <text position: B25> ]
<text in text symbol>
/* CIF TextEnd */

```

Cette règle est utilisée par <CIF descriptor: A18>.

```

<text in text symbol> ::=
chaîne de caractères (et non une construction PR)

```

Information complémentaire:

Un caractère de "Nouvelle ligne" placé avant ou après l'un des deux commentaires de texte CIF ne doit pas être considéré comme faisant partie du texte à afficher dans le symbole de texte.

Exemple:

```

/* CIF Text (800,550) */
Timer myTimer;
/* CIF TextEnd */

```

A69 Symbole de sélection <select symbol: A69>:

```

/* CIF Select <pointlist: B22> */
[ <text position: B25> ]
<beginning of select definition>

```

Cette règle est utilisée par <CIF descriptor: A18>.

```

<beginning of select definition> ::=
SELECT IF <boolean simple expression> <end>

```

Information complémentaire:

Les points constituant la liste décrivent tous les coins du symbole de sélection, soit dans le sens des aiguilles d'une montre, soit dans le sens inverse.

Exemple:

```
/* CIF Select (700,400),(1100,400),(1100,750),(700,750) */
/* CIF TextPosition (725,425) */
SELECT IF (p = 3);
```

A70 Fin de descripteur <descriptor end: A70>:

```
{ /* CIF End Decision */ <end of decision> |
/* CIF End State */ <part of end of state> |
/* CIF End Label */ <part of end of free action> |
/* CIF End Select */ <end of select definition> |
/* CIF End TransitionOption */ <end of transition option> }
```

Cette règle est utilisée par <CIF descriptor: A18>.

<end of decision> ::=

ENDDECISION

<part of end of state> ::=

ENDSTATE [<state name>] <end>

<part of end of free action> ::=

ENDCONNECTION [<connector name>] <end>

<end of select definition> ::=

ENDSELECT <end>

<end of transition option> ::=

ENDALTERNATIVE

Information complémentaire:

Cette règle a été introduite pour faire la distinction entre la terminaison d'une information liée à un symbole et la terminaison d'un diagramme.

6.4.2 Règles CIF de type B**B1 Parties de diagramme <diagram parts: B1>:**

```
{ <page declaration: B3>+ | <nested frame: B13> }
```

Cette règle est utilisée par <system type diagram start: A6>, <block diagram start: A7>, <process diagram start: A10>, <service diagram start: A12>, <service type diagram start: A13>, <procedure diagram start: A14>, <operator diagram start: A15>.

Information complémentaire:

Si le diagramme est graphiquement inclus dans le diagramme englobant, un commentaire <nested frame: B13> doit être utilisé.

Dans le cas contraire, une déclaration <page declaration: B3> doit être spécifiée pour chaque page du diagramme.

B2 Parties de diagrammes et références d'accès <diagram parts & gate references: B2>:

```
{ <page declaration: B3> [ <gate reference: B19>+ ] }+ |
{ <nested frame: B13> [ <gate reference: B19>+ ] }
```

Cette règle est utilisée par <block type diagram start: A8>, <substructure diagram start: A9>, <process type diagram start: A11>.

Information complémentaire:

Si le diagramme est graphiquement inclus dans le diagramme englobant, un commentaire <nested frame: B13> doit être utilisé.

Dans le cas contraire, une déclaration <page declaration: B3> doit être spécifiée pour chaque page du diagramme.

Les références <gate reference: B19> sont facultatives. Elles ne sont utilisées que pour spécifier les positions du texte pour des références d'accès attachées au cadre du diagramme et qui sont présentes sur la page ou le cadre imbriqué défini précédemment. Si une référence <gate reference: B19> est omise, la position du texte est calculée automatiquement. Dans le PR, le nom de la référence d'accès est indiqué en relation avec la partie de PR contenant les acheminements de signaux ou les canaux connectés.

Les références <gate reference: B19> ne peuvent pas apparaître dans des pages <ServicePage>.

B3 Déclaration de page <page declaration: B3>:

```
/* CIF <page type: B12> <page name> <size point: B26> */  
[ <frame declaration: B14> ]  
[ <diagram heading text position: B25> ]  
[ <page text position: B23> ]  
[ <package reference: B18> ]
```

Cette règle est utilisée par <package diagram start: A4>, <system diagram start: A5>, <macro diagram start: A16>, <diagram parts: B1>, <diagram parts & gate references: B2>.

Information complémentaire:

Si la déclaration <frame declaration: B14> n'est pas présente, la position du cadre est fixée à (0,0) et sa taille est fixée à celle de la page. Cette page devient la page courante et le reste tant que, soit une nouvelle déclaration <page declaration: B3>, soit un changement <page switch: A20>, n'est pas rencontré.

La position <page text position: B23> définit le coin supérieur *droit* du rectangle englobant le texte.

La référence <package reference: B18> ne peut exister que si le diagramme est un progiciel ou un diagramme de système.

La référence <package reference: B18> ne peut exister que si le symbole de référence de progiciel est visible dans le SDL-GR.

Exemple:

```
/* CIF Page 1 (1900,2300) */  
/* CIF Frame (100,250), (1700,1950) */
```

B4 Symbole de sous-structure de canal <channel substructure symbol: B4>:

```
/* CIF ChannelSubstructure <position and size: B24> */  
[ <text position: B25> ]  
<dashed line: B21>
```

Cette règle est utilisée par <channel: A21>.

Information complémentaire:

La liste des points décrit le symbole <dashed association symbol> connectant la sous-structure de canal au canal. Le premier point de la liste est situé sur le rectangle englobant le symbole de sous-structure de canal. Le dernier point de la liste est situé sur le canal connecté à la sous-structure de canal.

B5 Symbole de contrainte d'accès <gate constraint symbol: B5>:

<block symbol rectangle: B15> | <process symbol rectangle: B16> | <service symbol rectangle: B17>

Cette règle est utilisée par <gate: A23>.

B6 Position de texte pour la première liste de signaux <first signallist text position: B6>:

```
/* CIF TextPosition <point: B26> SignalList1 */
```

Cette règle est utilisée par <channel: A21>, <signal route: A22>, <gate: A23>.

Information complémentaire:

Cette position de texte correspond à la première liste de signaux mentionnée dans le PR.

Exemple:

```
/* CIF TextPosition (800,550) SignalList1 */
```

B7 Position de texte pour la seconde liste de signaux <second signallist text position: B7>:

```
/* CIF TextPosition <point: B26> SignalList2 */
```

Cette règle est utilisée par <channel: A21>, <signal route: A22>, <gate: A23>.

Information complémentaire:

Cette position de texte correspond à la seconde liste de signaux mentionnée dans le PR.

B8 Position de la première flèche <first arrow position: B8>:

```
/* CIF Arrow1Position <point: B26> */
```

Cette règle est utilisée par <channel: A21>.

Information complémentaire:

Ce commentaire définit la position de la flèche pour la direction mentionnée en premier dans le PR pour le canal <channel: A21>.

Exemple:

```
/* CIF Arrow1Position (800,550) */
```

B9 Position de la seconde flèche <second arrow position: B9>:

```
/* CIF Arrow2Position <point: B26> */
```

Cette règle est utilisée par <channel: A21>.

Information complémentaire:

Ce commentaire définit la position de la flèche pour la direction mentionnée en second dans le PR pour le canal <channel: A21>.

B10 Texte d'accès entrant <inlet text: B10>:

```
/* CIF InletText <macro label> */  
[ <text position: B25> ]
```

Cette règle est utilisée par <macro call symbol: A61>, <macro inlet symbol: A62>.

B11 Texte d'accès sortant <outlet text: B11>:

```
/* CIF OutletText <macro label> */  
[ <text position: B25> ]
```

Cette règle est utilisée par <macro call symbol: A61>, <macro outlet symbol: A63>.

B12 Type de page <page type: B12>:

Page | BlockPage | ServicePage

Cette règle est utilisée par <page declaration: B3>.

Information complémentaire:

Dans le cas normal, le mot clé **Page** doit être utilisé.

Le mot clé **BlockPage** est utilisé pour des diagrammes de blocs et de types de bloc, afin de faire la distinction entre des pages contenant des blocs et des pages contenant des processus. Si le mot clé **BlockPage** est présent, alors la page contient des blocs. Si le mot clé **Page** est présent, alors la page contient des processus.

Le mot clé **ServicePage** est utilisé pour des diagrammes de processus et de types de processus, afin de faire la distinction entre des pages contenant des transitions et des pages contenant des services. Si le mot clé **ServicePage** est présent, alors la page contient des services. Si le mot clé **Page** est présent, alors la page contient des transitions. Notons que toutes les pages d'un même diagramme de processus doivent être du même type.

B13 Cadre imbriqué <nested frame: B13>:

```
/* CIF NestedFrame <position and size: B24> */
[ <diagram heading text position: B25> ]
```

Cette règle est utilisée par <diagram parts: B1>, <diagram parts & gate references: B2>.

B14 Déclaration de cadre <frame declaration: B14>:

```
/* CIF Frame <position and size: B24> */
```

Cette règle est utilisée par <page declaration: B3>.

B15 Rectangle de symbole de bloc <block symbol rectangle: B15>:

```
/* CIF Block <position and size: B24> */
```

Cette règle est utilisée par <block symbol: A30>, <gate constraint symbol: B5>.

B16 Rectangle de symbole de processus <process symbol rectangle: B16>:

```
/* CIF Process <position and size: B24> */
```

Cette règle est utilisée par <process symbol: A32>, <gate constraint symbol: B5>.

B17 Rectangle de symbole de service <service symbol rectangle: B17>:

```
/* CIF Service <position and size: B24> */
```

Cette règle est utilisée par <service symbol: A34>, <gate constraint symbol: B5>.

B18 Référence de progiciel <package reference: B18>:

```
/* CIF PackageReference <position and size: B24> */
[ <text position: B25> ]
```

Cette règle est utilisée par <page declaration: B3>.

B19 Référence d'accès <gate reference: B19>:

```
/* CIF GateReference <connection point: B26> */
<text position: B25>
```

Cette règle est utilisée par <block symbol: A30>, <dashed block symbol: A31>, <process symbol: A32>, <dashed process symbol: A33>, <service symbol: A34>, <dashed service symbol: A35>, <diagram parts & gate references: B2>.

Information complémentaire:

Ce commentaire CIF sert à spécifier la position du texte d'une référence d'accès. Le point de connexion correspond au point où la référence d'accès se connecte au canal ou à l'acheminement de signaux. Le nom de l'accès est trouvé dans la partie de PR contenant le canal ou l'acheminement de signaux connecté. Pour en savoir plus sur les accès et les références d'accès, se reporter au module <channel: A21>. Voir aussi le chapitre 7 "Exemples". Ce commentaire ne doit être utilisé que pour des connexions à travers des accès. Dans le cas de connexions entre canaux et acheminements de signaux sans utilisation d'accès, il faut directement employer le module <connect: A24>.

Exemple:

```
/* CIF GateReference (800,550) */
/* CIF TextPosition (750,500) */
```

B20 Ligne <line: B20>:

```
/* CIF Line <pointlist: B22> */
```

Cette règle est utilisée par <text extension: A25>, <flow line: A28>, <answer flow line: A29>.

B21 Ligne en pointillé <dashed line: B21>:

```
/* CIF Line <pointlist: B22> Dashed */
```

Cette règle est utilisée par <comment: A26>, <channel substructure symbol: B4>.

B22 Liste de points <pointlist: B22>:

<first point: B26> { , <next point: B26> }+

Cette règle est utilisée par <channel: A21>, <signal route: A22>, <gate: A23>, <create line: A27>, <select symbol: A69>, <line: B20>, <dashed line: B21>.

B23 Position de texte de page <page text position: B23>:

/* CIF TextPosition <point: B26> **PageName** ***/**

Cette règle est utilisée par <page declaration: B3>.

B24 Position et taille <position and size: B24>:

<position point: B26> [, <size point: B26>]

Cette règle est utilisée par <text extension: A25>, <comment: A26>, <dashed block symbol: A31>, <dashed process symbol: A33>, <dashed service symbol: A35>, <system type symbol: A36>, <block type symbol: A37>, <process type symbol: A38>, <service type symbol: A39>, <block substructure symbol: A40>, <procedure symbol: A41>, <operator symbol: A42>, <start symbol: A43>, <stop symbol: A44>, <state symbol: A45>, <nextstate symbol: A46>, <save symbol: A47>, <task symbol: A48>, <set symbol: A49>, <reset symbol: A50>, <export symbol: A51>, <create symbol: A52>, <procedure call symbol: A53>, <procedure start symbol: A54>, <return symbol: A55>, <decision symbol: A56>, <continuous signal symbol: A57>, <enabling condition symbol: A58>, <transition option symbol: A59>, <join symbol: A60>, <macro call symbol: A61>, <macro inlet symbol: A62>, <macro outlet symbol: A63>, <label symbol: A64>, <input symbol: A65>, <priority input symbol: A66>, <output symbol: A67>, <text symbol: A68>, <nested frame: B13>, <frame declaration: B14>, <channel substructure symbol: B4>, <block symbol rectangle: B15>, <process symbol rectangle: B16>, <service symbol rectangle: B17>, <package reference: B18>.

Information complémentaire:

La taille <size point: B26> peut être omise si une taille a été définie par défaut avant au moyen du commentaire <default size: A19>. La position <position point: B26> correspond au coin supérieur *gauche* du rectangle englobant s'il n'y a rien de spécifié dans des commentaires placés avant.

B25 Position de texte <text position: B25>:

/* CIF TextPosition <point: B26> ***/**

Cette règle est utilisée par <channel: A21>, <signal route: A22>, <gate: A23>, <connect: A24>, <text extension: A25>, <comment: A26>, <answer flow line: A29>, <block symbol: A30>, <dashed block symbol: A31>, <process symbol: A32>, <dashed process symbol: A33>, <service symbol: A34>, <dashed service symbol: A35>, <system type symbol: A36>, <block type symbol: A37>, <process type symbol: A38>, <service type symbol: A39>, <block substructure symbol: A40>, <procedure symbol: A41>, <operator symbol: A42>, <start symbol: A43>, <state symbol: A45>, <nextstate symbol: A46>, <save symbol: A47>, <task symbol: A48>, <set symbol: A49>, <reset symbol: A50>, <export symbol: A51>, <create symbol: A52>, <procedure call symbol: A53>, <procedure start symbol: A54>, <return symbol: A55>, <decision symbol: A56>, <continuous signal symbol: A57>, <enabling condition symbol: A58>, <transition option symbol: A59>, <join symbol: A60>, <macro call symbol: A61>, <label symbol: A64>, <input symbol: A65>, <priority input symbol: A66>, <output symbol: A67>, <text symbol: A68>, <select symbol: A69>, <page declaration: B3>, <nested frame: B13>, <channel substructure symbol: B4>, <package reference: B18>, <gate reference: B19>, <inlet text: B10>, <outlet text: B11>.

Information complémentaire:

Ce point définit le coin supérieur *gauche* du rectangle englobant le texte s'il n'y a rien de spécifié dans des commentaires placés avant. La largeur et la hauteur du texte ne sont pas définies.

Exemple:

```
/* CIF TextPosition (800,550) */
```

B26 Point <point: B26>:

(<integer> , <integer>)

Cette règle est utilisée par <default size: A19>, <page declaration: B3>, <gate reference: B19>, <pointlist: B22>, <text position: B25>, <page text position: B23>, <first signallist text position: B6>, <second signallist text position: B7>, <first arrow position: B8>, <second arrow position: B9>, <position and size: B24>.

6.5 Commentaires CIF spécifiques aux outils

C0 commentaire CIF spécifique aux outils <tool-specific CIF comment: C0>:

```
/* CIF [ Keep ] Specific <name of tool> <tool-specific information> */
```

Information complémentaire:

Si le mot clé **Keep** est présent, ce commentaire CIF spécifique-outil doit être conservé dans l'objet CIF en cours d'édition. Dans le cas contraire, ce commentaire doit être supprimé lorsque l'objet CIF est édité.

Une règle spécifique-outil doit être associée à une règle CIF de type A (règle équivalant à un objet CIF). Elle doit être placée entre les commentaires CIF et les constructions SDL-PR associées à la règle A correspondante. L'ordre d'apparition des règles CIF spécifiques-outils pour une même règle de type A n'est pas fixé.

Exemple informel (incorrect par rapport aux grammaires CIF et SDL-PR):

```
/* CIF 'Ce commentaire CIF est associé à la règle Ax'*/  
/* CIF Specific mySDLTool 'Cette information est supportée par mySDLTool.  
Les outils autres que mySDLTool peuvent ou non comprendre cette  
information. Ce commentaire CIF spécifique-outil est lié à la règle Ax'*/  
'Ceci est le SDL-PR associé à la règle Ax';
```

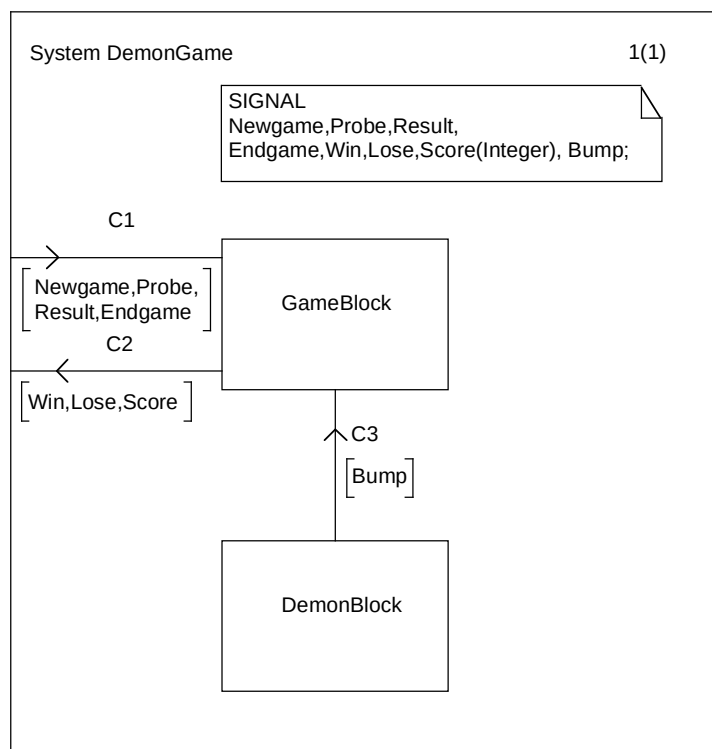
7 Exemples

Le présent paragraphe montre quelques exemples SDL-GR et leur correspondance en SDL-CIF. Trois diagrammes complets (mais simples) sont d'abord présentés. Puis, des constructions SDL-GR délicates à traiter sont présentées en détail.

Notons que la plupart des exemples présentés ici ne sont pas des diagrammes imbriqués, c'est-à-dire que le mot clé REFERENCED est utilisé dans le SDL-PR. Il est bien sûr aussi autorisé en SDL-CIF de créer des diagrammes imbriqués, pour lesquels le SDL-PR ne contient pas de mots clés REFERENCED.

7.1 Système du jeu DemonGame

7.1.1 Système du jeu DemonGame en SDL-GR



T1008990-96/d20

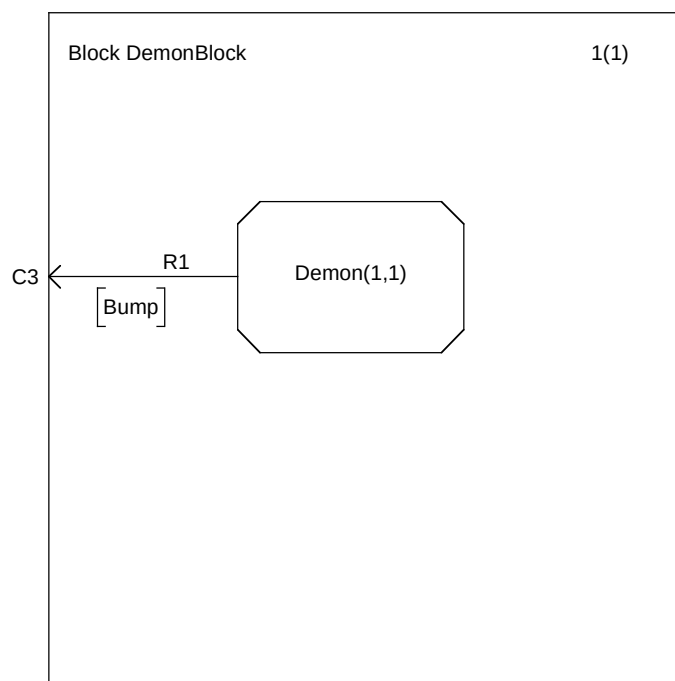
Système du jeu DemonGame en SDL-CIF

```

/* CIF SystemDiagram */
/* CIF Page 1 (1150,1000) */
System DemonGame;
/* CIF Specific mySDLTool Page 1 NoGrid */
/* CIF DefaultSize (300,200) */
/* CIF Text (400,100),(600,100) */
SIGNAL
Newgame,Probe,Result,
Endgame,Win,Lose,Score(Integer),Bump;
/* CIF TextEnd */
/* CIF Channel (550,700),(550,500) */
/* CIF TextPosition (575,551) */
/* CIF TextPosition (575,600) SignalList1 */
channel C3
from DemonBlock to GameBlock
with Bump;
endchannel C3;
/* CIF Channel (400,475),(0,475) */
/* CIF TextPosition (150,425) */
/* CIF TextPosition (62,500) SignalList1 */
channel C2
from GameBlock to env
with Win,Lose,Score;
endchannel C2;
/* CIF Channel (0,325),(400,325) */
/* CIF TextPosition (212,250) */
/* CIF TextPosition (50,350) SignalList1 */
channel C1
from env to GameBlock
with Newgame,Probe,
Result,Endgame;
endchannel C1;
/* CIF BlockSymbol (400,700) */
block DemonBlock referenced;
/* CIF BlockSymbol (400,300) */
block GameBlock referenced;
/* CIF End SystemDiagram */
endsystem DemonGame;

```

7.1.2 Bloc DemonBlock en SDL-GR



T1009000-96/d21

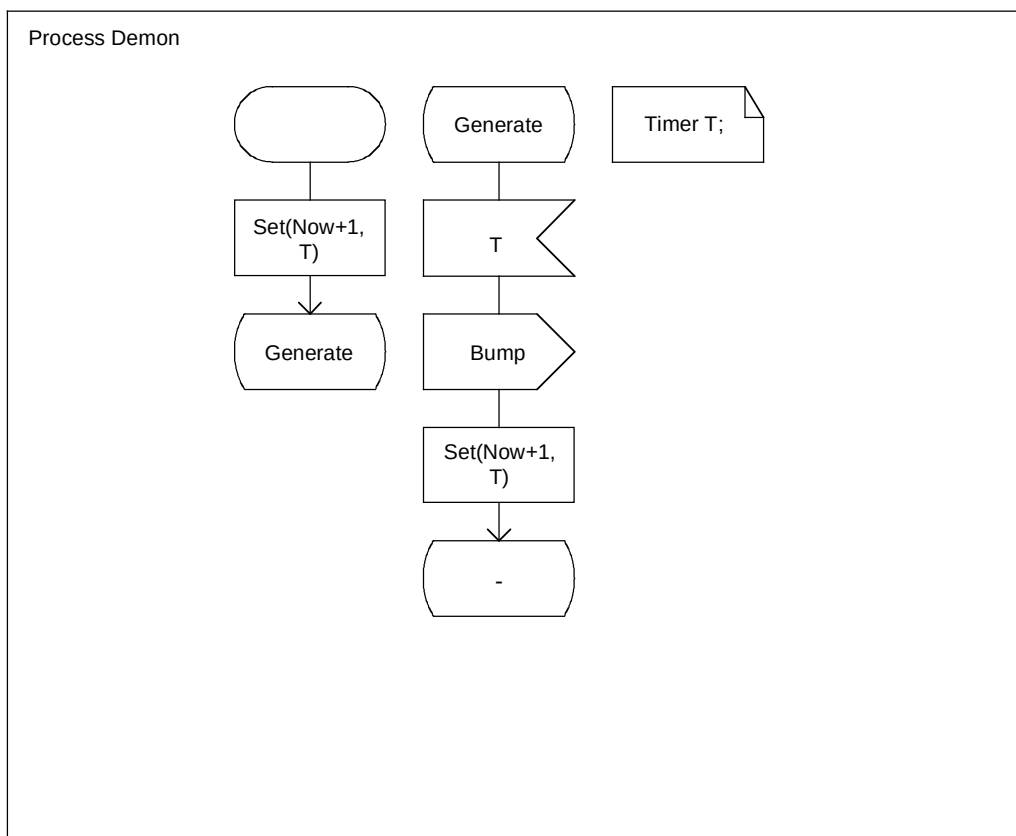
Bloc DemonBlock en SDL-CIF

```

/* CIF BlockDiagram */
/* CIF Page 1 (1000,1000) */
/* CIF Frame (100,100),(800,800) */
Block DemonBlock;
/* CIF Specific mySDLTool Page 1 NoGrid */
/* CIF DefaultSize (300,200) */
/* CIF Connect */
/* CIF TextPosition (25,300) */
Connect C3 and R1;
/* CIF SignalRoute (250,350),(0,350) */
/* CIF TextPosition (150,300) */
/* CIF TextPosition (75,375) SignalList1 */
signalroute R1
from Demon to env with Bump;
/* CIF Process (250,250) */
process Demon (1,1) referenced;
/* CIF End BlockDiagram */
endblock DemonBlock;

```

7.1.3 Processus Demon en SDL-GR



T1009010-96/d22

Processus Demon en SDL-CIF (sans spécification de lignes de flux)

```

/* CIF ProcessDiagram */
/* CIF Page 1 (1400,1000) */
Process Demon ;
/* CIF DefaultSize (200,100) */
/* CIF Text (800,100) */
Timer T;

```

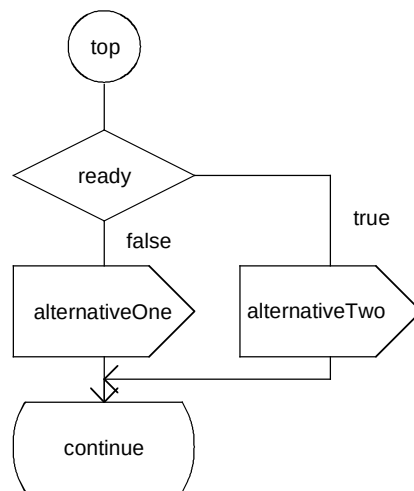
```

/* CIF TextEnd */
/* CIF Start (300,100) */
start ;
/* CIF Set (300,250) */
Set(Now+1,
T) ;
/* CIF NextState (300,400) */
nextstate Generate ;
/* CIF State (550,100) */
state Generate ;
/* CIF Input (550,250) */
input T ;
/* CIF Output (550,400) */
output Bump ;
/* CIF Set (550,550) */
Set(Now+1,
T) ;
/* CIF NextState (550,700) */
nextstate - ;
/* CIF End ProcessDiagram */
endprocess Demon;

```

7.2 Constructions SDL complexes

7.2.1 Jonction de lignes de flux, exemple 1



T1009020-96/d23

Notons que la liste des points pour la ligne de flux placée après le symbole de sortie alternativeTwo contient quatre points, deux d'entre eux étant placés sur le contour des symboles.

```

/* CIF Label (50,100), (100,100) */
top :
/* CIF Line (100,200), (100,300) */
/* CIF Decision (0,300) */
decision ready;
/* CIF Answer */
/* CIF Line (100,400), (100,500) */
(false) :
/* CIF Output (0,500) */
output alternativeOne;
/* CIF Line (100,600), (100,700) */
/* CIF Answer */

```

```

/* CIF Line (200,350),(400,350),(400,500) */
(true) :
  /* CIF Output (300,500) */
  output alternativeTwo;
/* CIF Line (400,600),(400,650),(100,650),(100,700) */
/* CIF End Decision */
enddecision;
/* CIF NextState (0,700) */
nextstate continue;

```

7.2.2 Jonction de lignes de flux, exemple 2

Cet exemple est similaire au précédent. C'est en fait le même SDL-GR, mais le PR est différent. Notons que, dans le PR, les deux lignes de flux se rejoignant sont placées à proximité de leurs symboles de départ.

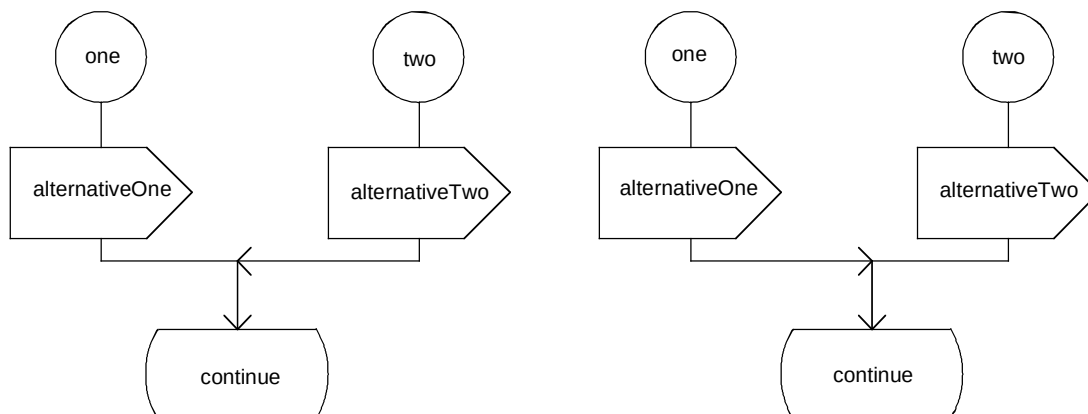
```

/* CIF Label (50,100),(100,100) */
top :
/* CIF Line (100,200),(100,300) */
/* CIF Decision (0,300) */
decision ready;
/* CIF Answer */
/* CIF Line (100,400),(100,500) */
(false) :
  /* CIF Output (0,500) */
  output alternativeOne;
  /* CIF Line (100,600),(100,700) */
  /* CIF Label Invisible */
  grs0 :
  /* CIF NextState (0,700) */
  nextstate continue;
/* CIF Answer */
/* CIF Line (200,350),(400,350),(400,500) */
(true) :
  /* CIF Output (300,500) */
  output alternativeTwo;
  /* CIF Line (400,600),(400,650),(100,650),(100,700) */
  /* CIF Join Invisible */
  join grs0;
/* CIF End Decision */
enddecision;

```

7.2.3 Jonction de lignes de flux, exemple 3

Le CIF ne précise pas quelle ligne de flux rejoint l'autre. Les deux exemples GR suivants peuvent produire le même SDL-CIF (l'exemple de PR présenté est sans lignes de flux).



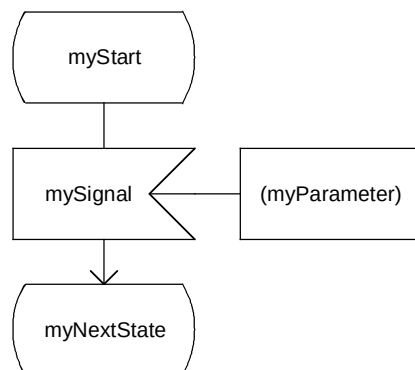
T1009030-96/d24

```

/* CIF Label (150,100),(100,100) */
connection
  one :
    /* CIF Output (100,300) */
    output alternativeOne;
    /* CIF Label Invisible */
    grs0 :
      /* CIF NextState (250,500) */
      nextstate continue;
/* CIF End Label */
endconnection one;
/* CIF Label (450,100),(100,100) */
connection
  two :
    /* CIF Output (400,300) */
    output alternativeTwo;
    /* CIF Join Invisible */
    join grs0;
/* CIF End Label */
endconnection two;

```

7.2.4 Lignes et rectangles englobants



T1009040-96/d25

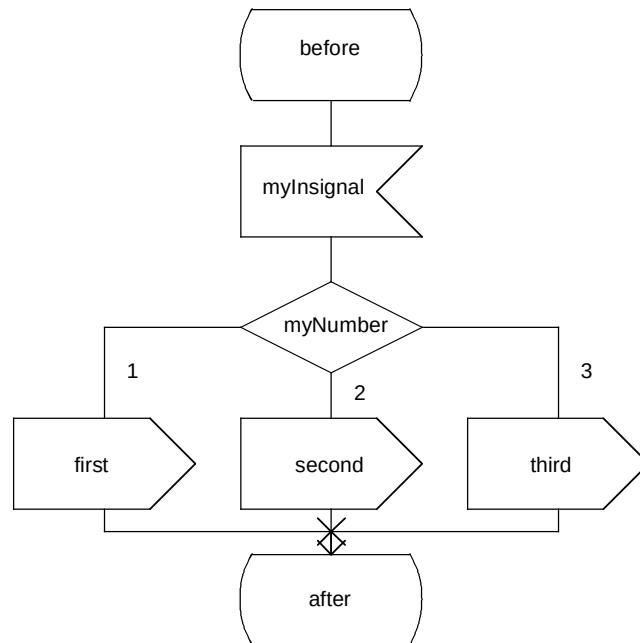
Notons que la ligne d'extension de texte est décrite comme si ses points terminaux étaient sur le bord des rectangles englobant les symboles associés, même si cela n'est pas complètement vrai dans le GR (le point terminal du symbole d'entrée est à l'intérieur du rectangle englobant). Cette règle s'applique à toutes les constructions de type lignes, acheminements de signaux et canaux, connectées à des symboles n'ayant pas une forme rectangulaire (symbole d'arrêt, symbole de processus...).

```

/* CIF State (100,100) */
state myStart;
  /* Input (100,300) */
  input mySignal
/* CIF TextExtension (400,300) */
/* CIF Line (400,350),(300,350) */
(myParameter)
/* CIF End TextExtension */
;
  /* CIF NextState (100,500) */
  nextstate myNextState;

```

7.2.5 Lignes de flux de réponse placées après une décision

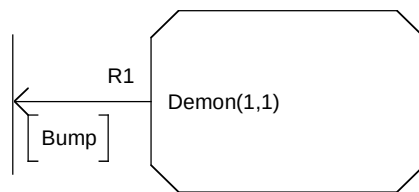


T1009050-96/d26

```
/* CIF State (400,100) */
state before;
/* CIF Input (400,300) */
input myInsignal;
/* CIF Decision (400,500) */
decision myNumber;
/* CIF Answer Left InvisibleBrackets */
(1) :
/* CIF Output (100,700) */
output First;
/* CIF Answer InvisibleBrackets */
(2) :
/* CIF Output (400,700) */
output second;
/* CIF Answer Right InvisibleBrackets */
(3) :
/* CIF Output (700,700) */
output third;
/* CIF End Decision */
enddecision;
/* CIF NextState (400,900) */
nextstate after;
/* CIF End State */
endstate;
```

7.2.6 Information de connexion en commentaire texte et en symboles rapportés sur le diagramme

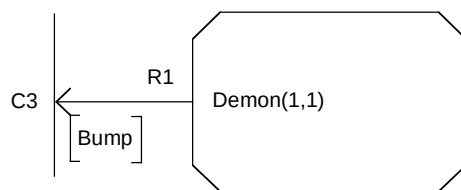
En GR, l'information de connexion peut être spécifiée de deux manières différentes mais produisant le même PR. En CIF, ces deux manières de préciser cette information sont distinguées par l'utilisation de commentaires CIF différents.



Connect C3 and R1;

T1009060-96/d27

```
/* CIF Text (200,800) */
Connect C3 and R1;
/* CIF TextEnd */
```

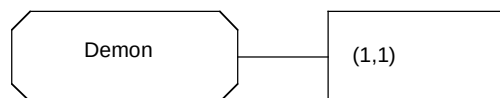


T1009060-96/d28

```
/* CIF Connect */
/* CIF TextPosition (25,700) */
Connect C3 and R1;
```

7.2.7 Extension de texte

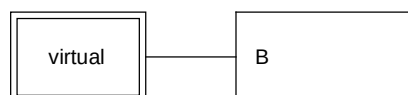
Ce diagramme GR est légal:



T1009070-96/d29

```
/* CIF Process (800,550) */
PROCESS Demon
/* CIF TextExtension (1100,550) */
(1,1)
/* CIF TextExtensionEnd */
REFERENCED;
```

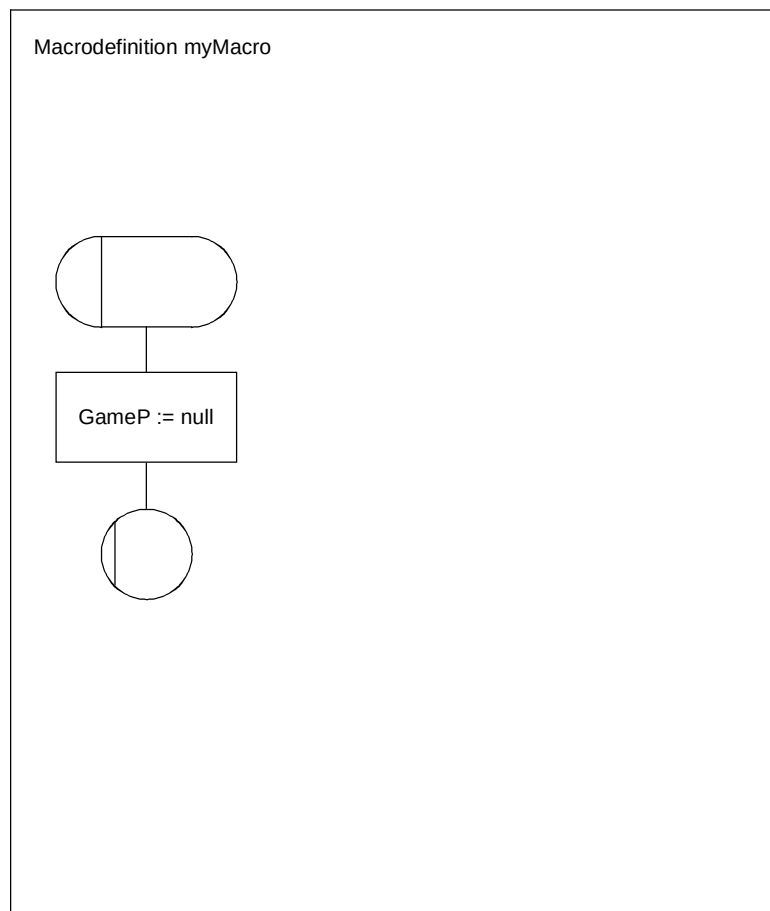
Ce diagramme GR est aussi légal:



T1009070-96/d30

```
/* CIF BlockType (800,550) */
virtual BLOCK TYPE
/* CIF TextExtension (1100,550) */
B
/* CIF TextExtensionEnd */
REFERENCED;
```

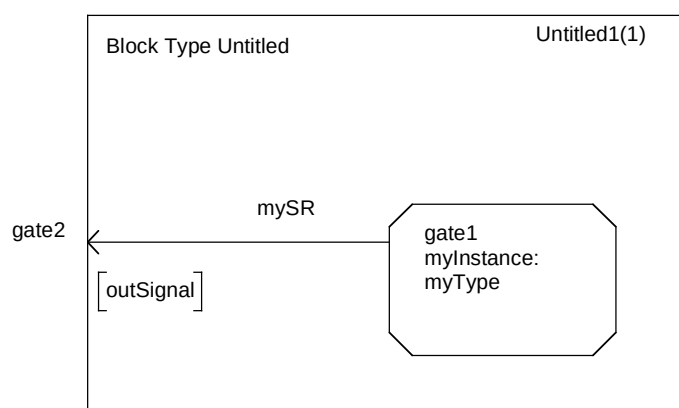
7.2.8 Diagramme de macro



T1009080-96/d31

```
/* CIF MacroDiagram */
/* CIF Page 1 (1000,1000) */
Macrodefinition myMacro;
/* CIF MacroInlet (300,100) */
/* CIF Task (300,300) */
TASK GameP := null;
/* CIF MacroOutlet (300,500) */
/* CIF End MacroDiagram */
ENDMACRO myMacro;
```

7.2.9 Position du texte pour des références d'accès



T1009090-96/d32

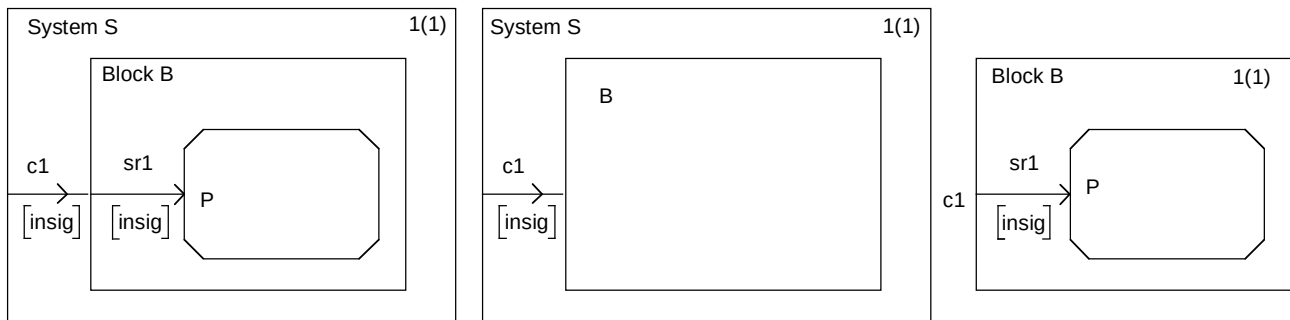
Notons que la position du texte pour l'accès gate2 est définie en relation avec la définition de la page ou du cadre, tandis que, dans le PR, le texte gate2 est défini pour l'acheminement de signal mySR. D'une manière similaire, la position du texte pour l'accès gate1 est définie en relation avec le symbole de processus myInstance, tandis que, dans le PR, le texte gate1 est défini pour l'acheminement de signal. Pour trouver à quel texte, la position de texte de la référence d'accès est associée, l'outil doit comparer les positions de la connexion de la référence d'accès avec les points terminaux de l'acheminement de signal.

```

/* CIF BlockTypeDiagram */
/* CIF Page Untitled1 (1900,1100) */
/* CIF Frame (100,100),(800,700) */
/* CIF GateReference (100,550) */
/* CIF TextPosition (10,540) */
Block Type Untitled;
/* CIF Process (300,500),(200,100) */
/* CIF GateReference (300,550) */
/* CIF TextPosition (310,540) */
PROCESS myInstance:
myType;
/* CIF SignalRoute (300,550),(100,550) */
/* CIF TextPosition (200,500) */
/* CIF TextPosition (130,570) SignalRoute1 */
SIGNALROUTE mySR
FROM myInstance VIA gate1 TO ENV VIA gate2 WITH outSignal;
/* CIF End BlockTypeDiagram */
ENDBLOCK TYPE;

```

7.2.10 Diagrammes imbriqués



T1009100-96/d33

Ce diagramme imbriqué est converti en deux diagrammes de la manière suivante:

- 1) Remplacer le diagramme de bloc dans le système S avec un symbole de référence de bloc.
- 2) Créer un diagramme de bloc séparé.
- 3) Insérer l'information indiquant à quel canal l'acheminement de signal est connecté.
- 4) Nommer la page avec un nom approprié.

a) *CIF pour lequel les diagrammes sont imbriqués*

```

/* CIF SystemDiagram */
/* CIF Page 1 (600,500) */
System S;
/* CIF Channel (0,250),(100,250) */
CHANNEL c1
FROM ENV TO B WITH insig;
/* CIF BlockDiagram */
/* CIF NestedFrame (100,100),(400,300) */
Block B;
/* CIF SignalRoute (100,250),(200,250) */
SIGNALROUTE sr1
FROM ENV TO P WITH insig;

```

```

/* CIF Connect */
CONNECT c1 AND sr1;
/* CIF Process (200,200),(200,100) */
PROCESS P REFERENCED;
/* CIF End BlockDiagram */
ENDBLOCK B;
/* CIF End SystemDiagram */
ENDSYSTEM S;

```

b) *CIF après suppression de l'imbrication*

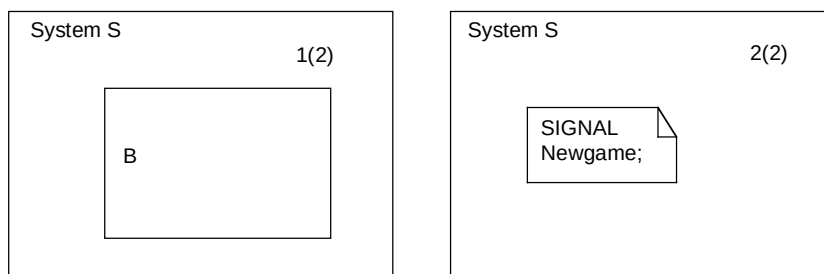
```

/* CIF SystemDiagram */
/* CIF Page 1 (600,500) */
System S;
/* CIF Channel (0,250),(100,250) */
CHANNEL c1
FROM ENV TO B WITH insig;
/* CIF Block (100,100),(400,300) */
BLOCK B REFERENCED;
/* CIF End SystemDiagram */
ENDSYSTEM S;

/* CIF BlockDiagram */
/* CIF Page 1 (600,500) */
/* CIF Frame (100,100),(400,300) */
Block B;
/* CIF SignalRoute (100,250),(200,250) */
SIGNALROUTE sr1
FROM ENV TO P WITH insig;
/* CIF Connect */
CONNECT c1 AND sr1;
/* CIF Process (200,200),(200,100) */
PROCESS P REFERENCED;
/* CIF End BlockDiagram */
ENDBLOCK B;

```

7.2.11 Diagramme en plusieurs pages



T1009110-96/d34

```

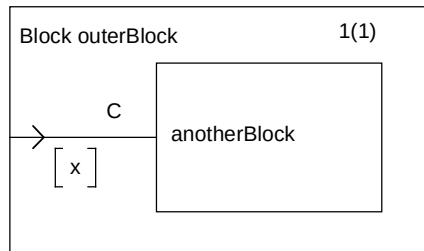
/* CIF SystemDiagram */
/* CIF Page 2 (600,300) */
/* CIF Page 1 (600,300) */
System S;
/* CIF DefaultSize (200,100) */
/* CIF Block (200,100) */
BLOCK B REFERENCED;
/* CIF CurrentPage 2 */
/* CIF Text (200,100) */
SIGNAL
NewGame;
/* CIF TextEnd */
/* CIF End SystemDiagram */
ENDSYSTEM S;

```

7.2.12 Bloc dans un bloc

Souvent en SDL-GR, des symboles de référence de bloc sont placés dans des diagrammes de bloc. Ce raccourci n'est pas autorisé en SDL-PR. Lorsque le PR est généré à partir du GR, le raccourci "bloc dans un bloc" est expansé au moyen d'une construction de sous-structure de bloc. Voici quelques exemples montrant comment le CIF traite cela, ainsi que des situations similaires. Les commentaires CIF correspondant au positionnement des textes ne sont pas précisés, afin de rendre le texte plus lisible.

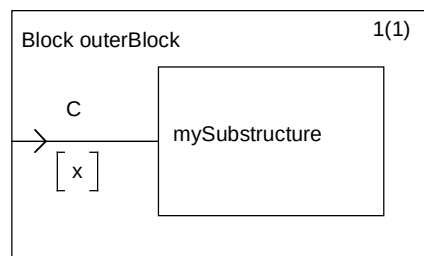
Symbole de référence de bloc dans un diagramme de bloc



T1009120-96/d35

```
/* CIF BlockDiagram */
/* CIF Page 1 (400,300) */
BLOCK outerBlock;
  /* CIF SubstructureDiagram Invisible */
  SUBSTRUCTURE outerBlock;
    /* CIF Channel (0,150),(100,150) */
    CHANNEL C FROM env TO anotherBlock WITH x;
    ENDCHANNEL C;
    /* CIF Block (100,100),(200,100) */
    BLOCK anotherBlock REFERENCED;
  /* CIF End SubstructureDiagram */
  ENDSUBSTRUCTURE outerBlock;
/* CIF End BlockDiagram */
ENDBLOCK;
```

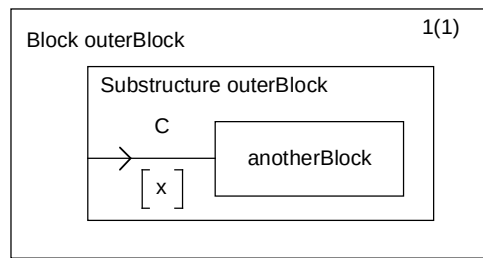
Symbole de référence de sous-structure dans un diagramme de bloc



T1009130-96/d36

```
/* CIF BlockDiagram */
/* CIF Page 1 (400,300) */
BLOCK outerBlock;
  /* CIF BlockSubstructure (100,100),(200,100) */
  SUBSTRUCTURE mySubstructure REFERENCED;
  /* CIF Channel (0,150),(100,150) */
  CHANNEL C FROM env TO anotherBlock WITH x;
  ENDCHANNEL C;
/* CIF End BlockDiagram */
ENDBLOCK;
```

Diagramme de sous-structure imbriqué dans un diagramme de bloc



T1009140-96/d37

Notons que le SDL-PR de cet exemple est exactement le même que le premier exemple vu dans ce sous-paragraphe, alors que le SDL-GR est différent.

```
/* CIF BlockDiagram */
/* CIF Page 1 (600,500) */
BLOCK outerBlock;
  /* CIF SubstructureDiagram */
  /* CIF NestedFrame (100,100),(400,300) */
  SUBSTRUCTURE outerBlock;
    /* CIF Channel (100,250),(200,250) */
    CHANNEL C FROM env TO anotherBlock WITH x;
    ENDCHANNEL C;
    /* CIF Block (200,200),(200,100) */
    BLOCK anotherBlock REFERENCED;
  /* CIF End SubstructureDiagram */
  ENDSUBSTRUCTURE outerBlock;
/* CIF End BlockDiagram */
ENDBLOCK;
```

7.3 Situations non traitées par le CIF



T1009150-96/d38

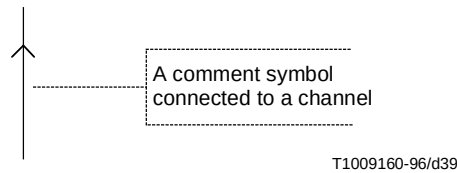
Le CIF ne sait pas traiter correctement ce diagramme. Le texte CIF correspondant aurait dû être:

```
/* CIF Process (800,550) Dashed myProcess */
```

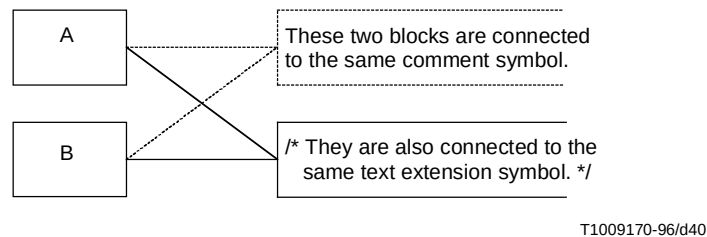
Le commentaire vu dans le GR sera perdu en CIF. La conséquence est que le CIF ne peut pas être utilisé comme format de stockage pour tous les diagrammes GR légaux. Pour traiter correctement cet exemple, le texte complet vu dans le symbole devrait être stocké dans le commentaire CIF associé. Cela voudrait dire que des caractères d'échappement auraient dû être introduits dans la grammaire CIF pour traiter le début de commentaire, fin de commentaire, début de texte et fin de texte. Cela n'a pas été fait afin d'éviter de trop compliquer la grammaire CIF.

L'exemple ci-dessus fait partie de cette catégorie de problèmes apparaissant dans le CIF dus au fait que le SDL-PR n'est pas totalement équivalent au SDL-GR. Des problèmes similaires pour introduire des commentaires textuels apparaissent, par exemple, avec les symboles de connexion, d'accès et de macro.

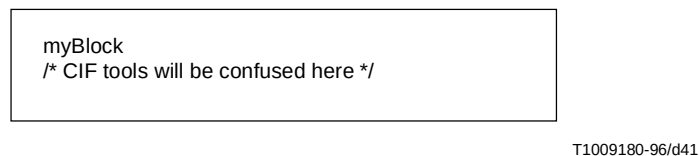
Voici un nouvel exemple de problème avec les commentaires. Cette fois, il s'agit d'un symbole de commentaire. Comment exprimer cela en PR?



Voici un nouvel exemple SDL-GR, qui peut être considéré comme légal ou non en SDL-GR, mais qui ne peut pas être traité correctement en CIF ou en PR (la conversion la plus fidèle, de ce GR en PR, serait de dupliquer les symboles de commentaire et d'extension; mais que se passerait-il si la traduction en PR devait être retraduite en GR?).



Voici une situation que le CIF ne peut pas traiter, due aux principes même du CIF. Cette situation correspond à une partie de GR contenant quelque chose de ressemblant à du CIF:



8 Critères de conformité au CIF

8.1 Lecture d'un fichier CIF

Un outil lisant un fichier CIF doit essayer de s'adapter plutôt que d'être trop strict vis-à-vis des règles CIF.

Exemple 1: le fichier CIF indique que la page courante est X, mais il n'y a pas de page X dans le diagramme courant. L'outil ne doit générer qu'un avertissement et ignorer ce commentaire CIF d'utilisation de page.

Exemple 2: le fichier CIF décrit une ligne de flux ne se connectant pas à des symboles. L'outil ne doit générer qu'un avertissement et ignorer ce commentaire CIF de description de ligne de flux.

8.2 Tracé automatique

Quand un fichier CIF est exporté par un outil, il contient des informations de positionnement liées au tracé des diagrammes SDL. Lorsqu'un autre outil importe ce fichier, il doit utiliser ces informations. Si au lieu de les utiliser, l'outil applique des algorithmes de tracé automatique, alors cet outil n'est conforme qu'au niveau 1 du CIF. Les outils supportant le CIF doivent clairement indiquer: s'ils appliquent un tracé automatique forcé lorsqu'ils chargent un fichier CIF, comment ils conservent l'information graphique et quel niveau de CIF est supporté par leurs fonctions d'importation et d'exportation.

8.3 Conservation et utilisation des informations spécifiques aux outils

Le CIF a été défini de manière à pouvoir inclure de l'information spécifique à un outil, cette information pouvant ou non être interprétée par d'autres outils. L'UIT-T recommande aux fabricants d'outils souhaitant développer des instructions propres à leurs outils, de rendre public ce jeu d'instructions afin d'éviter toute interférence avec d'autres outils. Un exemple a été fourni dans cette version de Recommandation, et les versions CIF futures pourront inclure des instructions spécifiques-outils qui permettront aux autres fabricants d'importer et de préserver ces instructions. Il n'est pas possible de se prévaloir de droit d'utilisation payant en ce qui concerne des instructions propres aux outils déclarées au moyen du mécanisme prévu dans la présente Recommandation.

Index des mots clés du CIF

Notons que cette table d'index n'inclut pas les commentaires CIF spécifiques aux outils.

L'ordre utilisé dans cette table d'index est basé sur la version anglaise ("*E*: ...")

A

Réponse (*E: Answer*) <A29: réponse de décision>

Position de la première flèche (*E: Arrow1Position*) <B8: position de la première flèche>

Position de la seconde flèche (*E: Arrow2Position*) <B9: position de la seconde flèche>

B

Bloc (*E: Block*) <block symbol: A30>, <B15: rectangle de symbole de bloc>

Diagramme de bloc (*E: BlockDiagram*) <A7: début de diagramme de bloc>, <A17: fin de diagramme>

Page de bloc (*E: BlockPage*) <B12: type de page>

Sous-structure de bloc (*E: BlockSubstructure*) <A40: symbole de sous-structure de bloc>

Type de bloc (*E: BlockType*) <A37: symbole de type de bloc>

Diagramme de type de bloc (*E: BlockTypeDiagram*) <A8: début de diagramme de type de bloc>, <A17: fin de diagramme>

C

Canal (*E: Channel*) <A21: canal>

Sous-structure de canal (*E: ChannelSubstructure*) <B4: symbole de sous-structure de canal>

Commentaire (*E: Comment*) <A26: commentaire>

Connexion (*E: Connect*) <A24: connexion>

Signal continu (*E: ContinuousSignal*) <A57: symbole de signal continu>

Création (*E: Create*) <A52: symbole de création>

Ligne de création (*E: CreateLine*) <A27: ligne de création>

Page courante (*E: CurrentPage*) <A20: changement de page>

D

Pointillé (*E: Dashed*) <A23: accès>, <A31: symbole de bloc en pointillé>, <A33: symbole de processus en pointillé>, <A35: symbole de service en pointillé>, <B21: ligne en pointillé>

Décision (*E: Decision*) <A56: symbole de décision>, <A70: fin de descripteur>

Taille par défaut (*E: DefaultSize*) <A19: taille par défaut>

E

Condition de validation (*E: EnablingCondition*) <A58: symbole de condition de validation>

Fin (*E: End*) <A5: début de diagramme de système>, <A17: fin de diagramme>, <A25: extension de texte>, <A68: symbole de texte>, <A70: fin de descripteur>

F

Cadre (*E: Frame*) <B14: déclaration de cadre>

G

Accès (*E: Gate*) <A23: accès>

Référence d'accès (*E: GateReference*) <B19: référence d'accès>

I

Texte d'accès entrant (*E: InletText*) <B10: texte d'accès entrant>

Entrée (*E: Input*) <A65: symbole d'entrée>

Invisible (*E: Invisible*) <A9: début de diagramme de sous-structure>, <A60: symbole de branchement>, <A64: symbole d'étiquette>

Crochets invisibles (*E: InvisibleBrackets*) <A29: réponse de décision>

J

Branchement (*E: Join*) <A60: symbole de branchement>

K

Conserver (*E: Keep*) <C0: commentaire CIF spécifique aux outils>

L

Étiquette (*E: Label*) <A64: symbole d'étiquette>, <A70: fin de descripteur>

Gauche (*E: Left*) <A25: extension de texte>, <A26: commentaire>, <A29: réponse de décision>, <A65: symbole d'entrée>, <A66: symbole d'entrée prioritaire>, <A67: symbole de sortie>

Ligne (*E: Line*) <line: B20>, <B21: ligne en pointillé>

M

Appel de macro (*E: MacroCall*) <A61: symbole d'appel de macro>

Diagramme de macro (*E: MacroDiagram*) <A16: début de diagramme de macro>, <A17: fin de diagramme>

Accès entrant de macro (*E: MacroInlet*) <A62: symbole d'accès entrant de macro>

Accès sortant de macro (*E: MacroOutlet*) <A63: symbole d'accès sortant de macro>

N

Cadre imbriqué (*E: NestedFrame*) <B13: cadre imbriqué>

Etat suivant (*E: NextState*) <A46: symbole d'état suivant>

O

Opérateur (*E: Operator*) <A42: symbole d'opérateur>

Diagramme d'opérateur (*E: OperatorDiagram*) <A15: début de diagramme d'opérateur>, <A17: fin de diagramme>

Texte d'accès sortant (*E: OutletText*) <B11: texte d'accès sortant>

Sortie (*E: Output*) <A67: symbole de sortie>

P

Diagramme de progiciel (*E: PackageDiagram*) <A4: début de diagramme de progiciel>, <A17: fin de diagramme>

Référence de progiciel (*E: PackageReference*) <B18: référence de progiciel>

Page (*E: Page*) <B12: type de page>

Nom de page (*E: PageName*) <B23: position de texte de page>

Entrée prioritaire (*E: PriorityInput*) <A66: symbole d'entrée prioritaire>

Procédure (*E: Procedure*) <A41: symbole de procédure>

Appel de procédure (*E: ProcedureCall*) <A53: symbole d'appel de procédure>

Diagramme de procédure (*E: ProcedureDiagram*) <A14: début de diagramme de procédure>, <A17: fin de diagramme>

Début de procédure (*E: ProcedureStart*) <A54: symbole de début de procédure>

Processus (*E: Process*) <A33: symbole de processus en pointillé>, <B16: rectangle de symbole de processus>

Diagramme de processus (*E: ProcessDiagram*) <A10: début de diagramme de processus>, <A17: fin de diagramme>

Type de processus (*E: ProcessType*) <A38: symbole de type de processus>

Diagramme de type de processus (*E: ProcessTypeDiagram*) <A11: début de diagramme de type de processus>, <A17: fin de diagramme>

R

Réinitialisation (*E: Reset*) <A50: symbole de réinitialisation>

Retour (*E: Return*) <A55: symbole de retour>

Droit (*E: Right*) <A25: extension de texte>, <A26: commentaire>, <A29: réponse de décision>, <A65: symbole d'entrée>, <A66: symbole d'entrée prioritaire>, <A67: symbole de sortie>

S

Sauvegarde (*E: Save*) <A47: symbole de sauvegarde>

Sélection (*E: Select*) <A69: symbole de sélection>, <A70: fin de descripteur>

Service (*E: Service*) <A35: symbole de service en pointillé>, <B17: rectangle de symbole de service>

Diagramme de service (*E: ServiceDiagram*) <A12: début de diagramme de service>, <A17: fin de diagramme>

Page de service (*E: ServicePage*) <B12: type de page>

Type de service (*E: ServiceType*) <A39: symbole de type de service>

Diagramme de type de service (*E: ServiceTypeDiagram*) <A13: début de diagramme de type de service>, <A17: fin de diagramme>

Armement (*E: Set*) <A49: symbole d'armement>

Première liste de signaux (*E: SignalList1*) <B6: position de texte pour la première ligne de signaux>

Seconde liste de signaux (*E: SignalList2*) <B7: position de texte pour la seconde ligne de signaux>

Acheminement de signal (*E: SignalRoute*) <A22: acheminement de signal>

Spécifique (*E: Specific*) <C0: commentaire CIF spécifique aux outils>
 Départ (*E: Start*) <A43: symbole de départ>
 Etat (*E: State*) <A45: symbole d'état>, <A70: fin de descripteur>
 Arrêt (*E: Stop*) <A44: symbole d'arrêt>
 Diagramme de sous-structure (*E: SubstructureDiagram*) <A9: début de diagramme de sous-structure>, <A17: fin de diagramme>
 Diagramme de système (*E: SystemDiagram*) <A5: début de diagramme de système>, <A17: fin de diagramme>
 Type de système (*E: SystemType*) <A36: symbole de type de système>
 Diagramme de type de système (*E: SystemTypeDiagram*) <A6: début de diagramme de type de système>, <A17: fin de diagramme>

T

Tâche (*E: Task*) <A48: symbole de tâche>
 Texte (*E: Text*) <A68: symbole de texte>
 Extension de texte (*E: TextExtension*) <A25: extension de texte>
 Fin d'extension de texte (*E: TextExtensionEnd*) <A25: extension de texte>
 Position de texte (*E: TextPosition*) <B6: position de la première liste de signaux>, <B7: position de la seconde liste de signaux>, <B23: position de texte de page>, <B25: position de texte>
 Option de transition (*E: TransitionOption*) <A59: symbole d'option de transition>, <A70: fin de descripteur>

Appendice I

Commentaires CIF spécifiques aux outils

I.1 Maintenance du CIF

Comme il est totalement impossible de prévoir dans la version de SDL-CIF normalisée ici toutes les requêtes qui pourraient émaner de fabricants d'outils en ce qui concerne des directives spécifiques aux outils, la syntaxe a été ouverte afin de pouvoir intégrer continûment de nouvelles directives (*directives spécifiques aux outils*).

Tout d'abord, le nom des outils doit être approuvé par la Commission d'études de l'UIT-T chargée de la Recommandation Z.106, afin de s'assurer qu'il n'y a pas de conflit dans les noms choisis.

Dans la perspective d'enrichir le SDL-CIF, les fabricants d'outils ayant créé de nouvelles directives spécifiques à leurs outils doivent les proposer à la Commission d'études de la Recommandation Z.106. Ainsi, ces nouvelles directives pourront être intégrées dans la présente Recommandation, en tant que directives: spécifiques-outils, obligatoires ou facultatives.

Les implémentations du SDL-CIF ayant réussi ou échoué devront être rapportées à la Commission d'études, afin d'ajuster la classification des directives en obligatoires ou facultatives.

Une liste des commentaires CIF spécifiques aux outils, et qui sont disponibles dans les outils du marché, sera mise à jour en continu. Elle sera accessible sur le site Web de l'UIT-T. En outre, le présent appendice présente une liste initiale des commentaires CIF spécifiques-outils qui sont connus lors de la rédaction de ce document.

I.2 Commentaires CIF spécifiques aux outils actuellement disponibles

Les commentaires CIF définis ci-après ne font pas partie de la norme CIF. Le présent appendice présente les commentaires spécifiques-outils qui sont actuellement disponibles dans les outils du marché.

La manière dont un outil doit traiter des commentaires spécifiques-outils est décrite dans le sous-paragraphe relatif aux commentaires CIF spécifiques aux outils.

Dans la suite du présent appendice, nous utiliserons un outil imaginaire dont le nom <tool name> est mySDLTool. Un développeur d'outil SDL-CIF devra substituer mySDLTool par le nom approprié de son outil, nom qui a été approuvé par la Commission d'études et qui identifie de manière unique son outil. Les mots clés propres aux outils et définis dans le présent appendice ne figurent pas dans l'index des mots clés.

I.2.1 Position des commentaires CIF spécifiques aux outils

Ce point a été abordé au sous-paragraphe 6.5 (règle C0), paragraphe "Information complémentaire".

Exemple:

Dans l'exemple ci-après, la majorité des commentaires CIF spécifiques-outils est associé à la règle CIF <system diagram: A5>. Des détails sur chacun de ces commentaires sont donnés plus loin.

```
/* CIF SystemDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Page 2 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
/* CIF Specific mySDLTool Version 1.0 */
/* CIF Specific mySDLTool Page 1 Scale 200 AutoNumbered
FixedHeadingSize (200,100) */
/* CIF Specific mySDLTool Page 2 Scale 200 AutoNumbered */
/* CIF Specific mySDLTool OriginalFileName 'mysystem.ssy' */
SYSTEM mySystem;
/* CIF CurrentPage 1 */
/* CIF Text (800,550),(200,100) */
/* CIF Specific mySDLTool FixedSize (200,100) */
dcl
    MyNo Integer;
timer
    DoorTimer;
/* CIF End Text */
...
```

C1 Numéro de version de l'outil <tool version number: C1>:

```
/* CIF Specific mySDLTool Version x.y */
<diagram description: A2>
```

Associée à:

Ce commentaire peut être associé à n'importe quelle règle décrivant la règle <diagram start: A3>.

Exemple (pour lequel cette règle est associée à la règle <system diagram start: A5>):

```
/* CIF SystemDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
/* CIF Specific mySDLTool Version 1.2 */
SYSTEM mySystem;
```

C2 Fichier d'origine <original file: C2>:

```
/* CIF Specific mySDLTool OriginalFileName <file name char string literal> */
<diagram start: A3>
```

Associée à:

Ce commentaire peut être associé à n'importe quelle règle décrivant la règle <diagram start: A3>.

Information complémentaire:

Ce commentaire est utilisé pour mémoriser le nom du fichier d'origine qui a été traduit en CIF. Quand le fichier CIF est retraduit dans le format d'origine, le même nom doit être réutilisé. Seul le nom du fichier (par exemple, a.ssy) doit être spécifié dans le littéral <file name char string literal>, et non le chemin complet (par exemple, /home/lat/a.ssy).

Exemple (pour lequel cette règle est associée à la règle <system diagram start: A5>):

```
/* CIF SystemDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
/* CIF Specific mySDLTool OriginalFileName 'mysystem.ssy' */
SYSTEM mySystem;
```

C3 Spécification de détails pour la page <page details specification: C3>:

```
/* CIF Specific mySDLTool Page <page number> [ ShowMeFirst ] [ Scale <integer> ] [ Grid <point: B26> ]
[ AutoNumbered ] [ FixedHeadingSize [ <fixed size point: B26> ] ] */
```

Associée à:

Ce commentaire peut être associé à n'importe quelle règle décrivant la règle <diagram start: A3>. Il peut y avoir zéro ou un commentaire de ce type par déclaration <page declaration: B3> dans le diagramme.

Information complémentaire:

Ce commentaire est utilisé afin de fournir des détails supplémentaires pour une page déjà définie:

- le mot clé **ShowMeFirst** doit être spécifié si la page doit s'afficher en premier plan lors du chargement du diagramme par l'outil. Dans le cas contraire, ce mot clé ne doit pas être utilisé;
- le mot clé **Scale** spécifie l'échelle utilisée par l'éditeur SDL pour afficher le diagramme. Cette échelle s'exprime sous la forme d'un pourcentage de la taille normale. Si aucune échelle n'est précisée, alors le diagramme est affiché dans sa taille normale (échelle fixée à 100%);
- le mot clé **Grid** contrôle la taille de la grille pour la page. Si ce mot clé n'est pas spécifié, alors la grille est fixée par défaut à (50,50), ce qui revient à ne pas avoir de grille;
- le mot clé **AutoNumbered** contrôle la numérotation automatique. S'il est utilisé, alors l'outil construira la numérotation des pages de la manière suivante: la première page déclarée **AutoNumbered** sera numérotée 1, la seconde page déclarée **AutoNumbered** sera numérotée 2, etc. Si l'utilisateur ajoute une nouvelle page déclarée **AutoNumbered** entre la première et la seconde, alors cette nouvelle page sera numérotée 2 et les suivantes seront décalées d'un. Un seul littéral <page name charstring literal> doit être spécifié dans une déclaration <page declaration: B3> pour la page, même si la page est déclarée **AutoNumbered**, cela afin d'être capable de référencer une page à partir d'autres commentaires CIF.

Le mot clé **FixedHeadingSize** indique que le symbole de déclaration de diagramme utilisé devra être d'une taille fixe et non adaptée au texte contenu. Le texte n'est alors plus entièrement visible. Pour voir le texte complet, l'utilisateur devra cliquer deux fois à l'intérieur du symbole de déclaration de diagramme. L'outil SDL pourra aussi utiliser cette taille réduite de symbole lorsque l'utilisateur aura masqué le texte par un double-clic.

Exemple (pour lequel cette règle est associée à la règle <system diagram start: A5>):

```
/* CIF SystemDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Page 2 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
/* CIF Specific mySDLTool Page 1 Scale 200 AutoNumbered */
/* CIF Specific mySDLTool Page 2 Scale 200 AutoNumbered */
SYSTEM mySystem;
/* CIF Text (800,550) */
SIGNAL Newgame;
/* CIF End Text */
```

C4 Taille fixe <fixed size: C4>:

```
/* CIF Specific mySDLTool FixedSize [ <fixed size point: B26> ] */
```

Associée à:

Ce commentaire est associé à la règle <text symbol: A68>.

Information complémentaire:

Ce commentaire permet de réduire la taille du symbole courant. Normalement, le commentaire CIF associé au symbole indique quelle taille sera utilisée pour afficher le texte. Avec ce nouveau commentaire, le symbole est réduit en taille. Le texte contenu n'est alors plus entièrement visible. Pour voir le texte complet, l'utilisateur devra cliquer deux fois à l'intérieur du symbole. L'outil SDL pourra aussi utiliser cette taille réduite de symbole lorsque l'utilisateur aura masqué le texte par un double-clic.

Exemple:

```
/* CIF Text (800,550),(200,200) */
/* CIF Specific mySDLTool FixedSize (200,100) */
dcl
    MyNo Integer;
timer
    DoorTimer;
/* CIF End Text */
```


SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	Maintenance: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux pour données et communication entre systèmes ouverts
Série Z	Langages de programmation