



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Z.106

(08/2002)

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

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

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

Recommandation UIT-T Z.106

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

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

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

Recommandation UIT-T Z.106

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

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*), (Rec. UIT-T Z.100). Le CIF a été défini pour permettre l'échange de spécifications SDL graphiques (SDL-GR, *graphical SDL specifications*) entre outils de fournisseurs différents n'utilisant pas le même format de stockage. Avant que le CIF n'ait été défini, la représentation textuelle du langage SDL (SDL-PR, *textual representation of SDL*) était utilisée pour échanger des spécifications, avec le défaut que, l'information graphique étant ignorée, une même spécification était 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 rend les utilisateurs plus indépendants des fournisseurs d'outils et permet aux organismes de normalisation d'approuver des spécifications au format SDL-CIF quel que soit l'outil qu'ils utilisent en interne. Il augmente aussi 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 à 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 la notation MSC. Le SDL-CIF a été conçu comme une extension du SDL. Le CIF est basé sur la syntaxe SDL-PR, représentation textuelle du SDL également définie dans la présente Recommandation. Le CIF peut être lu et écrit par des outils ainsi que par des utilisateurs. Toutes les constructions disponibles en SDL peuvent être exprimées sous forme graphique ou sous forme SDL-PR purement textuelle. Les contraintes applicables à la présentation graphique sont exprimées en CIF par l'adjonction d'annotations spécifiques au SDL-PR. En conséquence, la plupart des descriptions SDL-PR existantes sont des descriptions légales en SDL-CIF. Le SDL-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 ainsi que les concepts de directive obligatoire et de directive facultative.

Le langage SDL-PR est une variante possible, purement textuelle, du langage SDL. Avant 2002, il était publié dans le contexte de la Rec. UIT-T Z.100, mais comme cette notation est principalement utilisée pour la communication au sein des outils et entre ceux-ci, la définition a été transférée dans la présente Recommandation. Le SDL-PR, qui est au format CIF niveau 0, permet l'échange de descriptions en SDL syntaxiquement complètes, généralement sous la forme d'un fichier par système. La conformité au langage SDL-PR exige que le modèle soit conforme à la sémantique correspondante définie dans la Rec. UIT-T Z.100.

La présente Recommandation introduit deux nouveaux niveaux de SDL-CIF. Deux autres niveaux de conformité sont définis, le premier à un niveau SDL-PR plus progressiste et le second incluant l'information graphique. 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 dans l'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 admet des spécifications SDL incomplètes. Le niveau 2 inclut le niveau 1 et peut admettre presque toutes les informations graphiques des diagrammes SDL-GR. Une spécification CIF doit préciser avec lequel des deux niveaux la conformité est assurée. De même, les fabricants d'outils utilisant le CIF doivent aussi indiquer le niveau de CIF auquel ils se conforment dans les fonctions d'importation et d'exportation de leurs produits.

Source

La Recommandation Z.106 de l'UIT-T, révisée par la Commission d'études 17 (2001-2004) de l'UIT-T, a été approuvée le 6 août 2002 selon la procédure définie dans la Résolution 1 de l'AMNT.

AVANT-PROPOS

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

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

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution 1 de l'AMNT.

Dans certains secteurs des technologies de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

NOTE

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

DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un Membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

© UIT 2003

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

TABLE DES MATIÈRES

		Page
1	Domaine d'application	1
2	Références normatives.....	1
3	Abréviations.....	1
4	Notation utilisée dans la présente Recommandation	1
5	CIF niveau 0 (SDL/PR)	1
5.1	Principes généraux.....	1
5.2	Règles générales	3
5.3	Organisation des spécifications SDL.....	3
5.4	Concepts structurels.....	4
5.5	Agents.....	7
5.6	Communication	9
5.7	Comportement	10
5.8	Données	16
5.9	Définition de système générique	17
6	CIF niveau 1 (CIF-PR)	18
6.1	Principes généraux.....	18
6.2	Unités transférables des spécifications SDL	18
6.3	Syntaxe CIF-PR.....	18
7	CIF niveau 2 (CIF-GR).....	19
7.1	Principes généraux.....	19
7.2	Principes généraux de la représentation graphique	20
7.3	Règles lexicales du CIF-GR	22
7.4	Syntaxe CIF-GR: règles CIF de type A.....	23
7.5	Syntaxe CIF-GR – Règles CIF de type B.....	45
7.6	Commentaires CIF spécifiques aux outils.....	51
8	Exemples	51
8.1	Système du jeu DemonGame	52
8.2	Constructions SDL complexes	55
8.3	Situations non traitées par le CIF	59
9	Critères de conformité au CIF	60
9.1	Lecture d'un fichier CIF	60
9.2	Tracé automatique/tracé forcé	60
9.3	Conservation et utilisation des informations spécifiques aux outils	61

	Page
Appendice I – Commentaires CIF spécifiques aux outils.....	61
I.1 Maintenance du CIF	61
I.2 Commentaires CIF spécifiques aux outils actuellement disponibles	61

Introduction

Depuis un certain nombre d'années, le langage de description et de spécification (SDL, *specification and description language*) est de plus en plus largement utilisé, tant dans l'industrie que dans les organismes de normalisation pour l'élaboration de normes et de Recommandations. Alors que dans l'industrie le SDL est souvent employé dans un environnement basé sur un seul outil, le travail de normalisation exige plus souvent une intégration des spécifications SDL provenant de différentes organisations qui appliquent des outils variés. Cette exigence est d'ailleurs fréquente dans les projets internationaux.

Jusqu'à l'élaboration de la présente 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 la perte d'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 commun, la présente Recommandation répond à un besoin exprimé depuis longtemps, à savoir, échanger des spécifications SDL sans perte d'informations graphiques.

Recommandation UIT-T Z.106

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

1 Domaine d'application

La présente Recommandation définit le format d'échange commun (CIF, *common interchange format*) pour des spécifications écrites dans le langage de description et de spécification (SDL) [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. Même si le format CIF permet aux utilisateurs d'écrire des spécifications directement en CIF, il n'est guère adapté à cela mais doit plutôt être généré par une spécification SDL existante rédigée conformément à [1].

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. La référence à un document figurant dans la présente Recommandation ne donne pas à ce document en tant que tel le statut d'une Recommandation.

[1] Recommandation UIT-T Z.100 (2002), *SDL: langage de description et de spécification*.

3 Abréviations

La présente Recommandation utilise les abréviations suivantes:

CIF format d'échange commun (*common interchange format*)

SDL langage de description et de spécification (*specification and description language*)

4 Notation utilisée dans la présente Recommandation

Le métalangage défini au § 5.4.2 de [1] a été utilisé et complété (voir ci-après) en vue de définir les propriétés et les syntaxes du format CIF. Lorsque le premier mot d'un symbole non terminal est séparé des mots suivants par un signe deux points, ce symbole est une référence à une autre règle du format CIF. Les symboles non terminaux se réfèrent à des non terminaux de la Rec. UIT-T Z.100 ou à des non terminaux introduits dans la présente Recommandation.

5 CIF niveau 0 (SDL-PR)

5.1 Principes généraux

Le format CIF niveau 0 propose une forme syntaxique additionnelle susceptible d'être utilisée dans la représentation d'un système: la représentation textuelle (SDL-PR). Celle-ci peut être utilisée en lieu et place de la représentation graphique de [1], désignée "SDL-GR" dans le présent document. Toutes deux étant des représentations concrètes du même langage SDL, elles sont équivalentes, surtout en ce qui concerne la grammaire abstraite pour les concepts correspondants.

Les représentations SDL-GR et SDL-PR ont un ensemble commun, appelé grammaire textuelle commune.

Bien que le SDL puisse être écrit soit en SDL-PR, soit en SDL-GR, le langage a été conçu en sachant que le SDL-PR ne sera pas fréquemment utilisé pour des besoins tels que l'échange entre outils, même si le format d'échange commun spécifié dans la suite du présent document limite encore davantage l'utilisation du SDL-PR. La plupart des utilisateurs se servent du SDL-GR.

La Figure 1 montre la relation entre le SDL-PR, le SDL-GR, les grammaires concrètes et la grammaire abstraite.

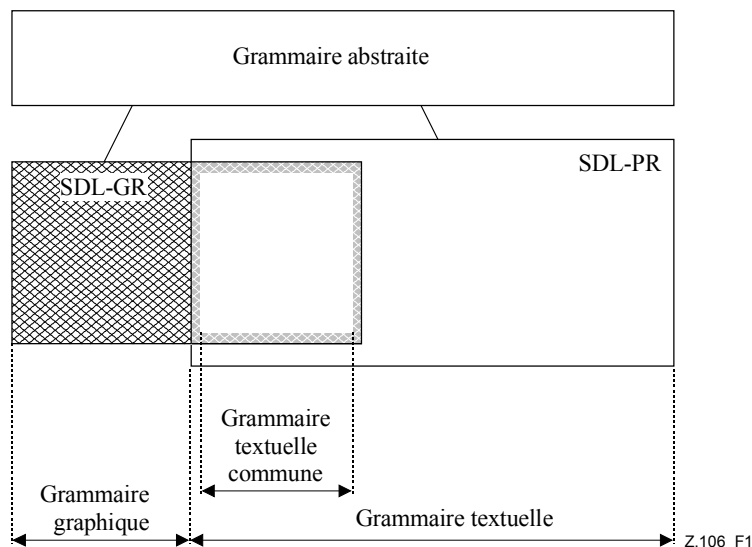


Figure 1/Z.106 – Grammaires du SDL

Chaque grammaire concrète a une définition de sa propre syntaxe et de sa relation avec la grammaire abstraite (autrement dit, comment opérer une transformation en syntaxe abstraite). Avec cette méthode, il n'y a qu'une seule définition de la sémantique du SDL; chacune des grammaires concrètes hérite de la sémantique via sa relation avec la grammaire abstraite. Cette approche garantit l'équivalence des langages SDL-PR et SDL-GR.

Pour certaines constructions, il n'y a pas de syntaxe abstraite directement équivalente. Dans ce cas, on propose un modèle pour la transformation de la syntaxe concrète en syntaxe concrète d'autres constructions qui ont (directement ou indirectement via d'autres modèles) une syntaxe abstraite. Les éléments qui n'ont pas de correspondance avec la syntaxe abstraite (telle que les commentaires) n'ont pas de signification formelle.

La syntaxe textuelle concrète pour le SDL-PR est spécifiée dans le formalisme de Backus-Naur élargi de la description de la syntaxe définie au § 5.4.2 de [1].

La syntaxe textuelle peut être suivie de paragraphes définissant les conditions statiques qui doivent être satisfaites dans un texte bien formé et qui peuvent être vérifiées sans interprétation d'une instance.

Dans de nombreux de cas, il existe une relation simple entre les syntaxes concrète et abstraite étant donné que la règle de syntaxe concrète est représentée simplement par une seule règle dans la syntaxe abstraite. Lorsque le même nom est utilisé dans les syntaxes abstraite et concrète pour indiquer qu'elles représentent le même concept, le texte "<x> de la syntaxe concrète représente X dans la syntaxe abstraite" est sous-entendu dans la description du langage et il est souvent omis. Dans le présent contexte, ce cas est ignoré mais les sous-catégories sémantiques soulignées (voir le § 5.4.2 de [1]) sont significatives.

Parallèlement aux productions pour la syntaxe concrète du SDL-PR, des paragraphes *Modèle* s'appliquent aux constructions qui sont de "syntaxe concrète dérivée" pour d'autres constructions de syntaxe concrète équivalentes (voir le § 5.3.2 de [1]).

NOTE – Dans le texte qui suit, les titres des sections sont alignés sur ceux des sections correspondantes de [1].

5.2 Règles générales

En langage SDL-PR, le nom ou l'identificateur optionnel qui, dans une définition, suit les mots clés de fin (**endsystem**, **endblock**, etc.) doit être syntaxiquement le même que le nom ou l'identificateur qui suit le mot clé de début correspondant (respectivement, **system**, **block**, etc.).

5.3 Organisation des spécifications SDL

5.3.1 Cadre d'application

La règle de production de départ de la Rec. UIT-T Z.100, <sdl spécification> (voir le § 7.1 de [1]), est remplacée par la production suivante:

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

Modèle

Une spécification <textual system specification> étant une définition <process definition> ou une définition <textual typebased process definition> est une syntaxe dérivée pour une définition <system definition> ayant le même nom que le processus, contenant des canaux implicites et contenant la définition <process definition> ou la <textual typebased process definition> comme définition unique.

Une spécification <textual system specification> étant une définition <block definition> ou une définition <textual typebased block definition> est une syntaxe dérivée pour une définition <system definition> ayant le même nom que le bloc, contenant des canaux implicites et contenant la définition <block definition> ou la définition <textual typebased block definition> comme définition unique.

Une clause <package use clause> précédant une <textual typebased agent definition> d'une <textual system specification> est une syntaxe dérivée pour une <package use clause> précédant le <system heading> de la <system definition> dérivée de la <textual typebased agent definition>.

5.3.2 Paquetage

```
<package definition>: :=  
    {<package use clause>*  
    <package heading> <end>  
    {<entity in package>*  
    endpackage [<package name>] <end>  
<entity in package>: :=  
    <agent type definition>  
    | <agent type reference>  
    | <package definition>  
    | <package reference>  
    | <signal definition>  
    | <signal reference>  
    | <signal list definition>  
    | <remote variable definition>
```

- | <textual data definition>
- | <data definition>
- | <data type reference>
- | <textual procedure definition>
- | <procedure definition>
- | <procedure reference>
- | <remote procedure definition>
- | <composite state type definition>
- | <composite state type reference>
- | <exception definition>
- | <select definition>
- | <macro definition>
- | <interface reference>
- | <association>

5.3.3 Définition référencée

La règle de production <referenced definition> (voir le § 7.3 de [1]) de la Rec. UIT-T Z.100 est remplacée par la production suivante:

<referenced definition>: :=
 <definition>

<definition>: :=
 <package definition>
 | <agent definition>
 | <agent type definition>
 | <composite state>
 | <composite state type definition>
 | <textual procedure definition>
 | <procedure definition>
 | <textual operation definition>
 | <operation definition>
 | <macro definition>

5.4 Concepts structurels

5.4.1 Types, instances et accès

5.4.1.1 Définitions de types structurels

5.4.1.1.1 Types d'agent

<agent type definition>: :=
 <system type definition> | <block type definition> | <process type definition>

5.4.1.1.2 Type de système

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

5.4.1.1.3 Type de bloc

<block type definition>: :=
 <package use clause>*
 <block type heading> <end> <agent structure>

endblock type [[<qualifier>] <block type name>] <end>

5.4.1.1.4 Type de processus

<process type definition>: :=
 <package use clause> *
 <process type heading> <end> <agent structure>
 endprocess type [[<qualifier>] <process type name>] <end>

5.4.1.2 Type d'état composite

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

5.4.1.2.1 Définitions fondées sur les types

<textual typebased agent definition>: :=
 <textual typebased system definition>
 | <textual typebased block definition>
 | <textual typebased process definition>

Le type d'agent désigné par le type <base type> dans l'expression de type d'une définition <textual typebased agent definition> doit contenir une transition de départ non étiquetée dans sa machine à états.

5.4.1.2.2 Définition de système fondée sur le type de système

<textual typebased system definition>: :=
 <typebased system heading> <end>

5.4.1.2.3 Définition de bloc fondée sur le type de bloc

<textual typebased block definition>: :=
 block <typebased block heading> <end>

5.4.1.2.4 Définition de processus fondée sur le type de processus

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

5.4.1.2.5 Définition d'état composite fondée sur le type d'état composite

<textual typebased state partition definition>: :=
 state aggregation <typebased state partition heading> <end>

5.4.1.3 Références de type

La règle de production <type referenced properties> (voir le § 8.1.5 de [1]) de la Rec. UIT-T Z.100 est remplacée par la production suivante:

<type reference properties>: :=
 [**with** { <attribute property> <end> } +]
 [**with** { <behaviour property> <end> } +]
 referenced <end>

5.4.1.4 Accès

<gate in definition>: :=
 <textual gate definition> | <textual interface gate definition>

<textual gate definition>: :=
gate <gate> [**adding**] <gate constraint> [<gate constraint>] <end>

<textual interface gate definition>: :=
gate { **in** | **out** } **with** <interface identifier> <end>

adding peut seulement être spécifié dans une définition de sous-type et seulement pour un accès défini dans le supertype. Lorsque **adding** est spécifié pour un accès <gate>, toutes les contraintes <textual endpoint constraint> et les listes <signal list> sont des adjonctions aux contraintes <gate constraint> de l'accès dans le supertype.

5.4.2 Associations

<association>: :=
association [<association name>] <association kind> <end>

<association kind>::=
 <association not bound kind>
 | <association end bound kind>
 | <association two ends bound kind>
 | <composition not bound kind>
 | <composition part end bound kind>
 | <composition composite end bound kind>
 | <composition two ends bound kind>
 | <aggregation not bound kind>
 | <aggregation part end bound kind>
 | <aggregation aggregate end bound kind>
 | <aggregation two ends bound kind>

<association not bound kind>: :=
from <association end> **from** <association end>

<association end bound kind>::=
from <association end> **to** <association end>

<association two ends bound kind>::=
to <association end> **to** <association end>

<composition not bound kind>::=
from <association end> **composition** <association end>

<composition part end bound kind>::=
to <association end> **composition** <association end>

<composition composite end bound kind>: :=
from <association end> **to composition** <association end>

<composition two ends bound kind>::=
to <association end> **to composition** <association end>

<aggregation not bound kind>::=
from <association end> **aggregation** <association end>

<aggregation part end bound kind>::=
to <association end> **aggregation** <association end>

<aggregation aggregate end bound kind>::=
from <association end> **to aggregation** <association end>

<aggregation two ends bound kind>::=
to <association end> **to aggregation** <association end>

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

Si une extrémité <association end> identifie un type d'agent ou une interface, la visibilité **protected** ne peut pas être utilisée dans l'autre extrémité <association end> de l'<association>.

Si deux <association> différentes identifient le même type, les noms <role name> (s'ils sont donnés) doivent être différents dans les extrémités <association end> opposées à ce type commun.

Il ne doit pas y avoir d'ensemble d'<association> contenant une composition telle que le type est relié à lui-même de par sa composition, que ce soit directement ou indirectement.

Si une extrémité <association end> est précédée du mot clé **composition** et identifie un type de données ou une interface, l'autre extrémité <association end> de l'<association> doit identifier respectivement un type de données ou une interface.

5.5 Agents

<agent definition>: :=
 <system definition> | <block definition> | <process definition>

<agent structure>: :=
 [<valid input signal set>]
 { { <entity in agent>
 | <channel to channel connection>
 | <channel definition>
 | <gate in definition>
 | <agent definition>
 | <agent reference>
 | <textual typebased agent definition> }*
 [<state partition>]
 | { <entity in agent>
 | <gate in definition> }*
 <agent body> }

La division <state partition> doit avoir le même nom que l'agent contenant. Elle définit la machine à états de l'agent. Si la structure <agent structure> peut être comprise à la fois comme division <state partition> et comme corps <agent body>, elle est interprétée comme corps <agent body>.

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

<entity in agent>: :=
 <signal definition>
 | <signal reference>
 | <signal list definition>
 | <variable definition>
 | <remote procedure definition>
 | <remote variable definition>
 | <textual data definition>
 | <data definition>
 | <data type reference>
 | <timer definition>
 | <interface reference>
 | <macro definition>
 | <exception definition>

- | <procedure reference>
- | <textual procedure definition>
- | <procedure definition>
- | <composite state>
- | <composite state type definition>
- | <composite state type reference>
- | <select definition>
- | <agent type definition>
- | <agent type reference>
- | <association>

<agent body>: :=
 [[<on exception>] <start>]
 { <state> | <exception handler> | <free action> } *

5.5.1 Système

<system definition>: :=
 {<package use clause>}*
 <system heading> <end> <agent structure>
endsystem [<system name>] <end>

5.5.2 Bloc

<block definition>: :=
 {<package use clause>}*
 <block heading> <end> <agent structure>
endblock [[<qualifier>] <block name>] <end>

5.5.3 Processus

<process definition>: :=
 {<package use clause>}*
 <process heading> <end> <agent structure>
endprocess [[<qualifier>] <process name>] <end>

5.5.4 Procédure

<textual procedure definition>: :=
 {<package use clause>}*
 <procedure heading> <end>
 <entity in procedure>*
 [<procedure body>]
endprocedure [[<qualifier>] <procedure name>] <end>

<entity in textual procedure>: :=
 <variable definition>
 | <textual data definition>
 | <data definition>
 | <data type reference>
 | <procedure reference>
 | <textual procedure definition>
 | <procedure definition>
 | <composite state type definition>
 | <composite state type reference>
 | <exception definition>
 | <select definition>

| <macro definition>

<textual procedure body>: :=
 [<on exception>] [<start>] { <state> | <exception handler> | <free action> } *

Une procédure **endprocedure**, contenue dans une définition <textual procedure definition> interne d'une procédure <entity in textual procedure> d'une définition <textual procedure definition> externe, appartient à cette définition <textual procedure definition> interne.

5.6 Communication

5.6.1 Canal

<channel definition>: :=
 channel [<channel name>] [**nodelay**]
 <channel path> [<channel path>]
 endchannel [<channel name>] <end>

<channel path>: :=
 from <channel endpoint>
 to <channel endpoint> [**with** <signal list>] <end>

<channel endpoint>: :=
 { <agent identifier> | <state identifier> | **env** | **this** } [<via gate>]

<via gate>: :=
 via <gate>

Le nom <channel name> de fin ne peut être spécifié que si le nom <channel name> de départ est spécifié. Si le nom <channel name> de départ n'est pas spécifié, le canal ne peut pas être désigné par un nom.

L'extrémité <channel endpoint> **this** désigne la machine à états de l'agent englobant directement la définition de canal.

L'accès <gate> doit être spécifié si:

- <channel endpoint> désigne une connexion à une définition <textual typebased agent definition> ou une <textual typebased state partition definition>, auquel cas l'accès <gate> doit être directement défini dans le type d'agent ou le type d'état pour cet agent ou cet état respectivement;
- env** est spécifié et le canal est défini dans un type d'agent, auquel cas l'accès <gate> doit être défini dans ce type d'agent.

Si l'accès <gate> est spécifié, le canal est relié à cet accès. L'accès et le canal doivent avoir au minimum un élément en commun dans leurs listes de signaux dans la même direction. Si aucun accès <gate> n'est spécifié, les règles suivantes s'appliquent: si l'extrémité de canal est un agent ou une machine à états et que cet agent/état contient une connexion <channel to channel connection> pour le canal, ce dernier est relié à l'accès implicite introduit par la connexion <channel to channel connection>. Autrement, si l'extrémité de canal est un état, le canal est relié à l'accès implicite de cette machine à états, le canal introduit un accès implicite sur l'agent ou l'état mentionné dans l'extrémité <channel endpoint>. Cet accès obtient la liste <signal list> des trajets <channel path> respectifs comme ses contraintes d'accès correspondantes. Le canal est relié à cet accès.

Si une définition <channel definition> contient deux <channel path>, les conditions suivantes doivent être satisfaites:

- le <channel endpoint> suivant **from** du premier <channel path> doit être le même que le <channel endpoint> suivant **to** du deuxième <channel path>;

- b) le <channel endpoint> suivant **to** du premier <channel path> doit être le même que le <channel endpoint> suivant **from** du deuxième <channel path>.

5.6.2 Connexion

<channel to channel connection>: :=
 connect <external channel identifiers>
 and <channel identifiers> <end>

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

Aucun canal ne peut être mentionné après le mot clé **and** dans plus d'une connexion <channel to channel connection> d'une unité de portée donnée.

Pour toute paire de connexions <channel to channel connection> d'une unité de portée donnée, les identificateurs <external channel identifiers> doivent mentionner le même ensemble de canaux ou ne doivent pas avoir de canal en commun. Si au moins deux connexions <channel to channel connection> d'une unité de portée donnée ont le même ensemble de canaux externes, ceci sert de syntaxe dérivée pour une connexion unique <channel to channel connection> qui utilise l'un des identificateurs <external channel identifiers> comme son identificateur <external channel identifiers> et ses identificateurs <channel identifiers> formés en énumérant tous les identificateurs internes <channel identifier>.

NOTE – A cause de l'élément **connect**, un canal (externe) pouvant être anonyme dans la version graphique d'une spécification, peut nécessiter l'attribution d'un nom dans la version textuelle correspondante. Ceci est rigoureusement analogue au cas des zones <merge area> dans les graphes de processus ou de procédure. Il convient donc qu'un outil permettant de convertir la version graphique d'une spécification en version textuelle puisse générer des noms de canaux implicites.

Modèle

Chaque connexion <channel to channel connection> différente dans une unité de portée donnée définit un accès implicite sur l'unité de portée. Tous les canaux dans la connexion <channel to channel connection> sont reliés à cet accès dans leurs unités de portée respectives. Les contraintes d'accès de l'accès implicite sont obtenues à partir des canaux reliés à l'accès.

Le nom de l'accès est un nom dérivé unique et non ambigu. Dans l'unité de portée qui l'entoure, la définition <channel definition> qui est définie par l'identificateur <channel identifier> est étendue avec une partie de l'accès <via gate>. La partie de l'accès <via gate> est ajoutée à l'extrémité <channel endpoint> qui référence l'unité de portée courante et elle mentionne l'accès implicite. A l'intérieur de l'unité de portée, les canaux qui sont associés au canal externe au moyen de la connexion <channel to channel connection> sont modifiés par extension de l'extrémité <channel endpoint> qui mentionne **env** avec une partie de l'accès <via gate> pour l'accès implicite.

5.7 Comportement

5.7.1 Départ

<start>: :=
 start [<virtuality>] [<state entry point name>] <end> [<on exception>] <transition>

Si le nom <state entry point name> est donné dans un départ <start>, le départ <start> doit être le départ <start> d'un état <composite state>.

5.7.2 Etat

<state>: :=
 state <state list> <end> [<on exception>]
 { <input part>

```

| <priority input>
| <save part>
| <spontaneous transition>
| <continuous signal>
| <connect part> } *
[ endstate [ <state name> ] <end> ]

```

Le nom <state name> optionnel se terminant par un état <state> ne peut être spécifié que si la liste <state list> dans l'état <state> consiste en un nom <state name> unique qui, dans ce cas, doit être ce nom <state name>.

La partie <connect part> n'est permise que pour un état <state> avec une liste <state list> qui contient un élément <composite state item>.

5.7.3 Entrée

```

<input part>: :=
    input [<virtuality>] <input list> <end>
    [<on exception>] [<enabling condition>] <transition>

```

5.7.4 Entrée prioritaire

```

<priority input>: :=
    priority input [<virtuality>]
    <priority input list> <end> [<on exception>] <transition>

```

5.7.5 Signal continu

```

<continuous signal>: :=
    provided [<virtuality>]
    <continuous expression> <end>
    [ priority <priority name> <end> ] [<on exception>] <transition>

```

5.7.6 Condition de validation

```

<enabling condition>: :=
    provided <provided expression> <end>

```

5.7.7 Sauvegarde

```

<save part>: :=
    save [<virtuality>] <save list> <end>

```

5.7.8 Transition spontanée

```

<spontaneous transition>: :=
    input [<virtuality>] <spontaneous designator> <end>
    [<on exception>] [<enabling condition>] <transition>

```

5.7.9 Etiquette

```

<label>: :=
    <connector name>:

```

NOTE – Dans la grammaire abstraite, seules les actions libres ont des étiquettes; les étiquettes à l'intérieur d'une transition sont transformées en actions libres séparées.

```

<free action>: :=
    connection
    <transition>
    [ endconnection [ <connector name> ] <end> ]

```

Si la chaîne <transition string> de la <transition> de l'action <free action> est non vide, la première instruction <action> doit avoir une étiquette <label>; autrement, l'instruction <terminator> doit avoir une étiquette <label>. Lorsqu'il est présent, le nom <connector name> terminant l'action <free action> doit être le même que le nom <connector name> dans cette étiquette <label>.

Modèle

Si une étiquette <label> n'est pas la première étiquette d'une chaîne <transition string>, la chaîne <transition string> est divisée en deux parties. Toutes les instructions <action> précédant l'étiquette <label> sont conservées dans la transition d'origine, qui est déterminée avec un branchement <join> à l'étiquette <label>. Toutes les instructions d'action suivant l'étiquette <label> sont copiées dans une nouvelle <free action>, qui débute avec l'étiquette <label>.

5.7.10 Machine à états et état composite

<composite state>: :=
 <composite state graph> | <state aggregation>

5.7.10.1 Graphe d'états composite

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

<composite state structure>: :=
 [<valid input signal set>]
 {<gate in definition>}*
 <state connection points>*
 <entity in composite state>*
 { <composite state body> | <state aggregation body> }

Une structure <composite state structure> ne doit contenir un corps <state aggregation body> que si elle est directement contenue dans une agrégation <state aggregation> ou une définition <composite state type definition> avec un en-tête <state aggregation type heading>; autrement, elle contient un <composite state body>. Une structure <composite state structure> qui contient un corps <state aggregation body> n'aura pas de <valid input signal set>.

<entity in composite state>: :=
 <variable definition>
 | <textual data definition>
 | <data definition>
 | <select definition>
 | <data type reference>
 | <macro definition>
 | <textual procedure definition>
 | <procedure definition>
 | <procedure reference>
 | <exception definition>
 | <composite state>
 | <composite state type definition>
 | <composite state type reference>

<composite state body>: :=
 [<on exception>] <start>* { <state> | <exception handler> | <free action> }*

5.7.10.2 Agrégat d'état

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

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

<state partition>: :=
 <textual typebased state partition definition>
 |
 <composite state reference>
 |
 <composite state>

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

<state partition connection>: :=
 connect <outer entry point> **and** <inner entry point> <end>

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

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

5.7.10.3 Point de connexion d'état

<state connection points>: :=
 { **in** <state entry points> | **out** <state exit points> } <end>

5.7.10.4 Connexion

<connect part>: :=
 connect [<virtuality>] [<connect list>] <end>
 [<on exception>] <exit transition>

<exit transition>: :=
 <transition>

5.7.11 Transition

5.7.11.1 Corps de transition

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

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

<action>: :=
 [<label>]
 {
 <task>
 |
 <output>
 |
 <create request>
 |
 <decision>
 |
 <set>
 |
 <reset>

```

| <export>
| <procedure call>
| <remote procedure call>
| <transition option> } <end>

```

```

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

```

Lorsque le terminateur <terminator> d'une transition <transition> est omis, la dernière action dans la <transition> doit contenir une <decision> terminale (voir le § 5.7.12.2) ou une option <transition option> terminale, sauf pour une <transition> contenue dans une <decision> ou une <transition option>.

5.7.11.2 Terminateur de transition

5.7.11.2.1 Etat suivant

```

<nextstate>: :=
    nextstate <nextstate body>

```

5.7.11.2.2 Branchement

```

<join>: :=
    join <connector name>

```

Un branchement <join> correspond à une zone <out connector area> en SDL-GR.

Il doit y avoir une seule et unique <connector name> correspondant à un branchement <join> à l'intérieur du même corps. La règle concernant le <agent body> dans une définition de type est énoncée au § 8.3.1 de [1].

5.7.11.2.3 Arrêt

```

<stop>: :=
    stop

```

5.7.11.2.4 Retour

```

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

```

5.7.11.2.5 Déclenchement d'exception (Raise)

```

<raise>: :=
    raise <raise body>

```

5.7.12 Action

5.7.12.1 Tâche

```

<task>: :=
    task [ <comment body> ] <left curly bracket> <task body> <right curly bracket>
    [ <end> <on exception> ]

```

5.7.12.2 Création

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

5.7.12.3 Appel de procédure

<procedure call>: :=
 call <procedure call body>

5.7.12.4 Sortie

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

5.7.12.5 Décision

<decision>: :=
 decision <question> <end> [<on exception>]
 <textual decision body>
 enddecision

<textual decision body>: :=
 <textual answer part>+ [<textual else part>]

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

<textual else part>: :=
 else <colon> [<transition>]

Une partie <textual answer part> ou une partie <textual else part> dans une décision ou une option de transition est une partie <textual answer part> terminale ou une partie <textual else part> terminale respectivement si elle contient une <transition> dans laquelle une instruction <terminator> est spécifiée, ou contient une chaîne <transition string> dont la dernière instruction <action> contient une décision terminale ou une option terminale. Une décision <decision> ou une option <transition option> est une décision terminale et une option terminale, respectivement, si chaque partie <textual answer part> et partie <textual else part> dans le corps <textual decision body> est une partie terminale <textual answer part> ou une partie terminale <textual else part>, respectivement.

La réponse <answer> de la partie <textual answer part> doit être omise si et seulement si la <question> est constituée par le mot clé **any**. Dans ce cas, il n'existe pas de partie <textual else part>.

Si une décision <decision> n'est pas une décision terminale, il s'agit d'une syntaxe dérivée pour une décision <decision> lorsque toutes les parties terminales <textual answer part> et la partie <textual else part> (si elle n'est pas terminale) ont inséré à l'extrémité de leur <transition> un branchement <join> vers la première instruction <action> qui suit la décision ou (si la décision est la dernière instruction <action> dans une chaîne <transition string>) vers l'instruction <terminator> suivante.

5.7.13 Temporisateur

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

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

Une clause <reset> représente un nœud *Reset-node*; une clause <set> représente un nœud *Set-node*.

5.7.14 Exception

5.7.14.1 Gestionnaire d'exception

<exception handler>: :=
 exceptionhandler <exception handler list> <end>
 [<on exception>]
 <handle>*
 [**endexceptionhandler** [<exception handler name>] <end>]

5.7.14.2 On-Exception

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

5.7.14.3 Traitement

<handle>: :=
 handle [<virtuality>] <exception stimulus list> <end>
 [<on exception>] <transition>

5.8 Données

5.8.1 Définitions de données

<textual data definition>: :=
 <textual data type definition>
 | <textual interface definition>
 | <textual syntype definition>

5.8.1.1 Définition des types de données

<textual data type definition>: :=
 {<package use clause>}*
 <type preamble> <data type heading> [<data type specialization>]
 <end> [<data type definition body> <data type closing> <end>]

<data type closing>: :=
 { **endvalue** | **endobject** } **type** [<data type name>]

Le mot clé **value** ou **object** et le nom <data type name> dans la fermeture <data type closing> doivent correspondre aux mots clés **endvalue** ou **endobject**, respectivement, et au nom dans l'en-tête correspondant <data type heading>, s'ils sont présents.

5.8.1.2 Définition d'interface

<textual interface definition>: :=
 {<package use clause>}*
 [<virtuality>] <interface heading>
 [<interface specialization>]
 <end> <entity in interface>* [<interface use list>]
 <interface closing>
 | {<package use clause>}*
 [<virtuality>] <interface heading>
 [<interface specialization>] <end>

<interface closing>: :=
 endinterface [<interface name>] <end>

5.8.1.3 Comportement des opérations

<textual operation definition>: :=
 {<package use clause>}*
 <operation heading> <end>
 <entity in textual operation>*
 <operation body>
 { **endoperator** | **endmethod** } [[<qualifier>] <operation name>] <end>

<entity in textual operation>: :=
 <textual data definition>
 | <data definition>
 | <variable definition>
 | <exception definition>
 | <select definition>
 | <macro definition>

<operation body>: :=
 [<on exception>] <start> { <free action> | <exception handler> }*

5.8.1.4 Syntypes

<textual syntype definition>: :=
 {<package use clause>}*
 syntype <syntype name> <equals sign> <parent sort identifier>
 { { <default initialization> [[<end>] [<constraint>] | <constraint> } <end>]
 <syntype closing>
 | [<syntype closing>] } <end>
 | {<package use clause>}*
 <type preamble> <data type heading> [<data type specialization>] <end>
 <data type definition body> <constraint> <end> <data type closing> <end>

<syntype closing>: :=
 endsyntype [<syntype name>]

5.9 Définition de système générique

5.9.1 Définition optionnelle

<select definition>: :=
 select if (<Boolean simple expression>) <end>
 {
 <agent type definition>
 | <agent type reference>
 | <agent definition>
 | <agent reference>
 | <channel definition>
 | <signal definition>
 | <signal list definition>
 | <signal reference>
 | <remote variable definition>
 | <remote procedure definition>
 | <textual data definition>
 | <data definition>
 | <data type reference>
 | <interface reference>
 | <composite state type definition>

```

| <composite state type reference>
| <state partition>
| <timer definition>
| <variable definition>
| <textual procedure definition>
| <procedure definition>
| <procedure reference>
| <channel to channel connection>
| <select definition>
| <macro definition>
| <exception definition> }+
endselect <end>

```

5.9.2 Chaîne de transition optionnelle

```

<transition option>: :=
    alternative <alternative question> <end>
    <textual decision body>
endalternative

```

6 CIF niveau 1 (CIF-PR)

6.1 Principes généraux

Le CIF niveau 1 est un surensemble de la syntaxe du SDL textuel (SDL-PR): il n'apporte pas d'informations complémentaires quant à la représentation graphique, il améliore seulement la capacité du SDL-PR en tant que format d'échange.

La principale limitation du SDL-PR en tant que format d'échange est qu'il n'autorise l'échange que de spécifications SDL complètes. Or, il y a un réel besoin d'échanger des descriptions partielles ou non encore achevées. Néanmoins, les descriptions doivent être syntaxiquement correctes pour pouvoir être échangées.

Le SDL-CIF utilise une grande partie de la syntaxe SDL-PR. Les règles de production du SDL-PR réutilisées dans le SDL-CIF ne sont que référencées par leur nom, elles ne sont pas décrites de nouveau dans la présente Recommandation.

Les règles de la Rec. UIT-T Z.100 définissant la grammaire abstraite du SDL-PR s'appliquent aussi aux parties du SDL-CIF communes avec le SDL-PR, dans la mesure du possible en fonction du caractère incomplet des descriptions SDL-CIF.

6.2 Unités transférables des spécifications SDL

6.3 Syntaxe CIF-PR

6.3.1 Fichier CIF

La règle de production initiale du langage SDL-PR, <sdl specification> (voir le § 5.3.1), est remplacée par la règle suivante:

```

<cif level 1 file>: :=
    { <textual system specification>
      | <definition>
    } *

```

6.3.2 Appel de macro

Dans le SDL, les appels de macro peuvent "apparaître partout où une unité <lexical unit> est autorisée" (§ 6.2.3 de [1]).

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

La règle de production <tâche> (voir le § 5.7.12.1) est alors remplacée par:

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

7 CIF niveau 2 (CIF-GR)

7.1 Principes généraux

Le CIF niveau 2 est une extension du CIF niveau 1 avec ce qu'il est convenu d'appeler les "directives CIF". Celles-ci permettent de décrire les principales caractéristiques de la représentation graphique des objets.

Les directives CIF sont placées avant l'objet auquel elles sont associées. L'un des principes de conception de la présente Recommandation a été le suivant: 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.

La première directive CIF d'un objet 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 réduirait considérablement le nombre d'outils capables de prendre en charge le SDL-CIF: 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 lignes 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 la présentation de texte à l'intérieur de symboles. Si une information facultative n'est pas fournie, alors l'outil détermine automatiquement une représentation.

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 ne peuvent être analysées que par leurs outils.

7.2 Principes généraux de la représentation graphique

7.2.1 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.

7.2.2 Pages

Les diagrammes peuvent être découpés en plusieurs pages, comme cela est décrit au § 6.6 de [1].

Toutefois, les fichiers SDL-CIF ne sont pas structurés en pages, mais en fonction de la syntaxe SDL-PR. Les pages sont décrites par des commentaires CIF insérés entre des unités syntaxiques. Une page peut contenir des informations provenant 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.

7.2.3 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"; par exemple, les canaux, les lignes de flux de transition et les lignes d'association;
- information relative aux symboles qui n'ont pas la forme de ligne, appelés par la suite "symboles"; par exemple, les symboles de processus ou les symboles de sortie;
- information relative au texte;
- autre information, par exemple celle relative à la découpe en pages.

7.2.4 Représentation des symboles

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 existe quelques exceptions lorsque c'est le coin supérieur droit qui sert de référence (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 aux symboles sont parfois ajoutées. Par exemple, pour les symboles existant dans les versions gauche et droite à la fois, les mots clés "Gauche" et "Droite" peuvent être ajoutés.

7.2.5 Représentation 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.

7.2.6 Directives facultatives pour la position du 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 en CIF. 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 de l'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 procédure, symbole d'opérateur, état, sauvegarde, tâche, armement, réinitialisation, création, appel de procédure, démarrage 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 paquetage.

7.2.7 Représentation de lignes

Le tracé de ligne est défini par une liste de coordonnées: celles du point de départ de la ligne, celles des éléments intermédiaires spécifient les points de rupture successifs de la ligne et celles du point d'arrivée de la ligne.

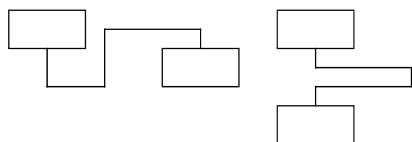
Généralement, les points de départ et d'arrivée sont connectés à d'autres symboles.

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 des canaux est une information obligatoire, étant donné qu'il est impossible de deviner ce que l'utilisateur veut réellement voir.

7.2.8 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-dessus.



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

Z.106_F2

Figure 2/Z.106

7.2.9 Informations graphiques non couvertes par le CIF

Les directives CIF sont principalement associées aux positions et aux tailles 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; 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.

7.2.10 Imbrication de diagrammes

L'imbrication de diagrammes SDL est prise en charge par le CIF. Un outil ne prenant pas en charge les diagrammes SDL imbriqués peut les convertir en symbole de référence et en diagrammes séparés:

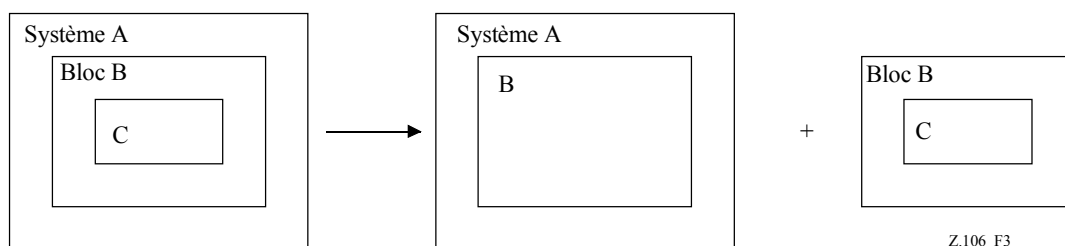


Figure 3/Z.106

7.2.11 Déclarations obligatoires et complémentaires de diagramme

Le CIF ne prend pas en charge les déclarations complémentaires de diagramme. Cela signifie que les outils prenant en charge les déclarations complémentaires 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. A noter que le SDL-GR de l'exemple ci-après n'est pas correct, étant donné que, d'après la Rec. UIT-T Z.100, il ne doit pas avoir de symbole entourant le texte de la déclaration complémentaire.

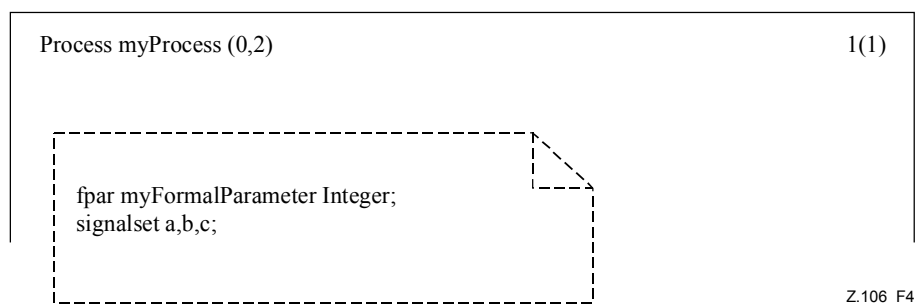


Figure 4/Z.106

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

7.3 Règles lexicales du CIF-GR

7.3.1 Directives CIF

Les directives CIF sont des formes spéciales de note <note> de la Rec. UIT-T 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 être une directive <cif directive>.

7.3.2 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 sont donc 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 faisant partie de l'objet texte.

Cela permet aux outils de prendre en compte les préférences de l'utilisateur en matière de présentation du texte.

Lorsque deux marques SDL d'un objet texte sont séparées par des caractères "Espace" et une nouvelle ligne suivie de 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 précédente directive CIF.

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

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

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").

7.3.3 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 SDL-GR sur deux lignes doit être stockée en SDL-PR comme suit:

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

au lieu de:

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

7.4 Syntaxe CIF-GR: règles CIF de type A

Les règles CIF de type A décrivent les directives qui sont associées à différentes constructions SDL-PR. En général, il y a une règle pour chaque construction SDL-PR qui correspond à un symbole graphique. Toutefois, dans certains cas un symbole est décrit par plusieurs règles étant donné que l'information dans le SDL-PR n'est pas rassemblée en un seul endroit. C'est le cas par exemple d'un cadre de diagramme où le début et la fin du diagramme sont décrits par des règles distinctes.

Les règles CIF de type B sont des règles utilitaires auxquelles renvoient les règles A et qui ne correspondent pas directement aux symboles dans les diagrammes.

Une règle normale CIF de type A 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 la spécification SDL-PR;
- une section composée d'un texte explicatif et d'un exemple.

A noter 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 niveau 2 n'est correct que s'il est conforme à la fois à la grammaire CIF niveau 2 (pour la définition des commentaires CIF) et à la grammaire CIF niveau 1.

7.4.1 A1 Description CIF <CIF description: A1>

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

Informations complémentaires:

cette règle définit la structure d'un fichier valide CIF niveau 2. Il convient de noter que cette règle ne correspond à aucun symbole du diagramme.

7.4.2 A2 Description diagramme <diagram description: A2>

```
<diagram start:A3> { <CIF descriptor:A19> | <page switch:A21> }* <diagram end:A18>
```

Informations complémentaires:

cette règle ainsi que celles qui concernent le début des diagrammes, <page switch:A21> et la règle <diagram end:A18> spécifient les cadres et les pages des diagrammes.

7.4.3 A3 Début de diagramme <diagram start: A3>

```
<specification area start: A5B> | <package diagram start: A5> | <system diagram start: A6> |  
<system type diagram start: A7> | <block diagram start: A8> | <block type diagram start: A9> |  
<substructure diagram start: A9> | <process diagram start: A10> | <process type diagram start:  
A11> | <state diagram start: A12> | <state type diagram start: A13> | <state aggregation diagram  
start: A14> | <state aggregation type diagram start: A15> | <procedure diagram start: A16> |  
<operator diagram start: A17>
```

7.4.4 A4 Début de zone de spécification <specification area start: A4>

```
/* CIF SpecificationArea */  
{ <page declaration: B2> }+
```

Informations complémentaires:

Il doit y avoir une déclaration <page declaration: B2> pour chaque page du diagramme.

7.4.5 A5 Début de diagramme de paquetage <package diagram start: A5>

```
/* CIF PackageDiagram */  
<diagram parts:B1>  
{<package use clause>}* <package heading> <end>
```

Exemple:

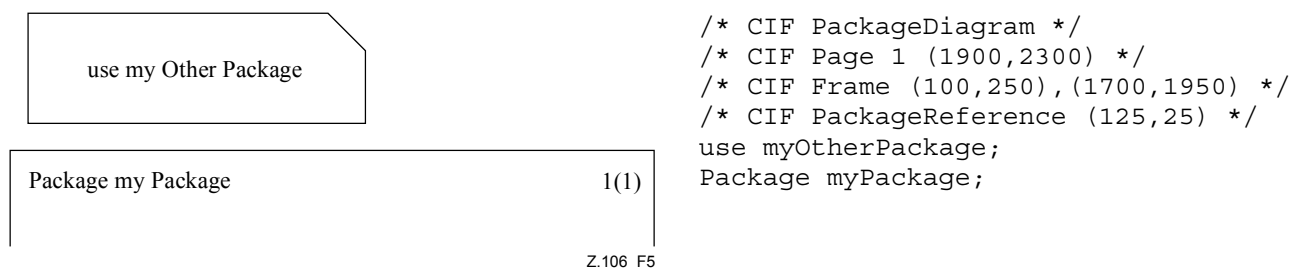


Figure 5/Z.106

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

```
/* CIF SystemDiagram */
{ <diagram parts: B1> }+
{ <package use clause> }*
<system heading> <end>
```

7.4.7 A7 Début de diagramme type de système <system type diagram start: A7>

```
/* CIF SystemTypeDiagram */
<diagram parts: B1>
<package use clause>* <system type heading> <end>
```

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

```
/* CIF BlockDiagram */<diagram parts: B1>
{<package use clause>}* <block heading>
```

7.4.9 A9 Début de diagramme type de bloc <block type diagram start: A9>

```
/* CIF BlockTypeDiagram */
<diagram parts: B1>
<package use clause>* <block type heading> <end>
```

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

```
/* CIF ProcessDiagram */
<diagram parts: B1>
{<package use clause>}* <process heading> <end>
```

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

```
/* CIF ProcessTypeDiagram */
<diagram parts: B2>
<package use clause>* <process type heading> <end>
```

7.4.12 A12 Début de diagramme d'état <state diagram start: A12>

```
/* CIF StateDiagram */
<diagram parts: B1>
{<package use clause>}* <composite state heading> <end>
```

7.4.13 A13 Début de diagramme type d'état <state type diagram start: A13>

```
/* CIF StateTypeDiagram */:
<diagram parts: B1>
{<package use clause>}* <composite state type heading> <end>
```

7.4.14 A14 Début de diagramme d'agrégation d'état <state aggregation diagram start: A14>

```
/* CIF StateAggregationDiagram */
<diagram parts: B1>
{<package use clause>}* <state aggregation heading> <end> substructure
```

7.4.15 A15 Début de diagramme type d'agrégation d'état < state aggregation type diagram start: A15

```
/* CIF StateAggregationTypeDiagram */
<diagram parts: B1>
{<package use clause>}* <state aggregation type heading> <end> substructure
```

7.4.16 A16 Début de diagramme de procédure <procedure diagram start: A16>

```
/* CIF ProcedureDiagram */
<diagram parts: B1>
{<package use clause>}* <procedure heading> <end>
```

7.4.17 A17 Début de diagramme d'opérateur <operator diagram start: A17>

```
/* CIF OperatorDiagram */
<diagram parts: B1>
{<package use clause>}* <operation heading> <end>
```

7.4.18 A18 Fin de diagramme <diagram end: A18>

```
{ /* CIF End SpecificationArea */ |
/* CIF End PackageDiagram */ ENDPACKAGE [ <package name> ] <end> |
/* CIF End SystemDiagram */ ENDSYSTEM [ <system name> ] <end> |
/* CIF End SystemTypeDiagram */ ENDSYSTEM TYPE [[<qualifier>] <system type name> ]
<end> |
/* CIF End BlockDiagram */ ENDBLOCK [[<qualifier>] <block name> ] <end> |
/* CIF End BlockTypeDiagram */ ENDBLOCK TYPE [[<qualifier>] <block type name> ]
<end> |
/* CIF End ProcessDiagram ENDPROCESS [[<qualifier>] <process name>] <end> |
/* CIF End ProcessTypeDiagram */ ENDPROCESS TYPE [[<qualifier>] <process type name>
] <end> |
/* CIF End StateDiagram */ ENDSUBSTRUCTURE [[<qualifier>] <composite state name>]
<end> |
/* CIF End StateTypeDiagram */ ENDSUBSTRUCTURE STATE TYPE [[<qualifier>]
<composite state type name>] <end> |
/* CIF End StateAggregationDiagram */ ENDSUBSTRUCTURE [[<qualifier>] <composite
state name>] <end> |
/* CIF End StateAggregationTypeDiagram */ ENDSUBSTRUCTURE STATE TYPE
[[<qualifier>] <composite state type name>] <end> |
/* CIF End ProcedureDiagram */ ENDPROCEDURE [[<qualifier>] <procedure name> ] <end>
|
/* CIF End OperatorDiagram */ ENDOPERATOR [[<qualifier>] <operator name> ] <end>
```

Exemple:

```
/* CIF End SystemTypeDiagram */
ENDSYSTEM TYPE mySystemType;
```

7.4.19 A19 Descripteur CIF <CIF descriptor: A19>

<diagram description: A2> | <default size: A20> | <channel: A22> | <gate: A23> | <gate symbol reference: A24> | <connect: A25> | <state connection point: A26> | <state connection: A27> | <text extension: A28> | <comment: A29> | <create line: A30> | <flow line: A31> | <answer flow line: A32> | <block symbol: A33> | <dashed block symbol: A34> | <process symbol: A35> | <dashed process symbol: A36> | <agent reference in specification area: A37> | <package symbol: A38> | <package reference in specification area: A39> | <operator symbol: A40> | <start symbol: A41> | <stop symbol: A42> | <state symbol: A43> | <nextstate symbol: A44> | <exception handler symbol: A45> | <on exception symbol: A46> | <handle symbol: A47> | <save symbol: A48> | <task symbol: A49> | <set symbol: A50> | <reset symbol: A51> | <export symbol: A52> | <create symbol: A53> | <procedure call symbol: A54> | <procedure start symbol: A55> | <return symbol: A56> | <raise symbol: A57> | <decision symbol: A58> | <continuous signal symbol: A59> | <enabling condition symbol: A60> | <transition option symbol: A61> | <join symbol: A62> | <connect symbol: A63> | <label symbol: A64> | <input symbol: A65> | <priority input symbol: A66> | <output symbol: A67> | <text symbol: A68> |

<select symbol: A69> | <descriptor end: A70> | <type reference: A71> | <association: A72> |
<specialization line: A73> | <dependency line: A74>

Informations complémentaires:

<diagram description: A2> est utilisé pour décrire les diagrammes imbriqués.

7.4.20 A20 Taille par défaut <default size: A20>

```
/* CIF DefaultSize <size point: B22> */
```

Informations complémentaires:

ce commentaire peut être placé avant n'importe quel symbole ou construction de ligne SDL-PR à l'intérieur d'un diagramme.

La taille indiquée ici sera utilisée pour tous les symboles subséquents sans une taille définie jusqu'à ce qu'une nouvelle taille par défaut soit spécifiée. La taille par défaut est rappelée après un nouveau <diagram start: A3>. En l'absence de taille par défaut, on ne peut pas omettre de spécifier de taille de symbole.

Exemple: cas où sera attribué au symbole de tâche la dimension (200, 100):

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

7.4.21 A21 Commutation de page <page switch: A21>

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

Informations complémentaires:

ce commentaire peut être placé avant un symbole ou une construction de ligne SDL-CIF et SDL-PR dans un diagramme. La page indiquée devient la page courante. Le nom de page doit renvoyer à une page déclarée dans le commentaire CIF de début de page précédent.

Tout ce qui suit ce commentaire CIF dans ce diagramme sera placé sur la page courante jusqu'à ce qu'une autre page courante soit définie. La page courante pour un diagramme est spécifiée par une déclaration <page declaration: B2>.

Exemple:

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

7.4.22 A22 Canal <channel: A22>

```
/* CIF Channel <pointlist: B18> [ InvisibleName ] */  
[ <channel name text position: B21> ]  
[ <first signallist text position: B4> ]  
[ <second signallist text position: B5> ]  
[ <first arrow position: B6> ]  
[ <second arrow position: B7> ]  
<channel definition>
```

Informations complémentaires:

si **InvisibleName** est spécifié, le nom de canal ne doit pas apparaître dans le SDL-GR. Le nom est indiqué dans le SDL-PR pour pouvoir se référer à lui dans une déclaration **CONNECT**.

Le premier point de la liste de points se trouve sur le rectangle entourant le symbole correspondant au **FROM** <channel endpoint> du premier <channel path> ou sur un accès connecté à ce symbole. Le dernier point de la liste de points se trouve sur le rectangle entourant ce symbole correspondant au **TO** <channel endpoint> du premier <channel path> ou sur un accès connecté à ce symbole.

La position du texte de l'accès dans la construction **VIA** est spécifiée dans <gate: A23> ou <gate reference: B15>. Le dessin ci-dessous représente un accès et une référence à cet accès.

Exemple 1:

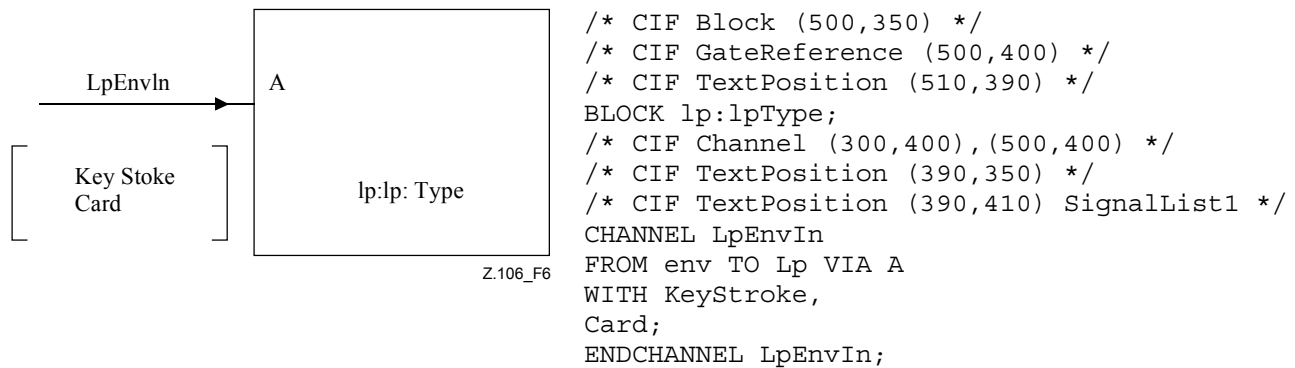


Figure 6/Z.106

7.4.23 A23 Accès <gate: A23>

```

/* CIF Gate <pointlist: B18> [ Dashed ] */
[ <gate name text position: B21> ]
[ <first signallist text position: B4> ]
[ <second signallist text position: B5> ]
[ <gate constraint symbol: B3> ]
<gate definition>

```

Informations complémentaires:

il doit y avoir deux points dans la liste <pointlist: B18>. Le premier point de la liste de points se trouve sur le rectangle englobant du symbole de cadre du diagramme. Le second point doit être l'autre point qui définit l'accès. Si une contrainte <textual endpoint constraint> existe comme faisant partie de la définition <gate definition>, le second point se trouvera sur le rectangle englobant du symbole correspondant à la contrainte <textual endpoint constraint>.

Dashed doit être utilisé si le mot clé **ADDING** est utilisé dans la définition <gate definition>.

Le symbole <gate constraint symbol: B3> doit être indiqué si une contrainte <textual endpoint constraint> est indiquée.

Exemple 1:

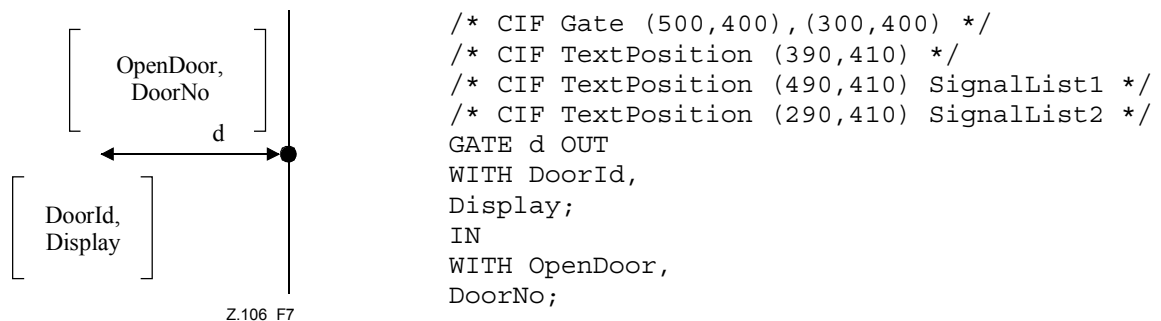


Figure 7/Z.106

Exemple 2:

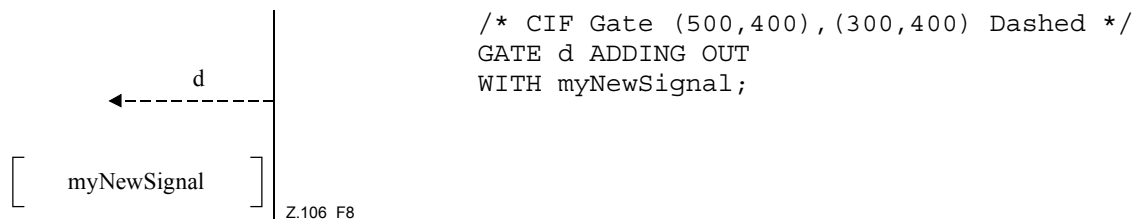


Figure 8/Z.106

Exemple 3:

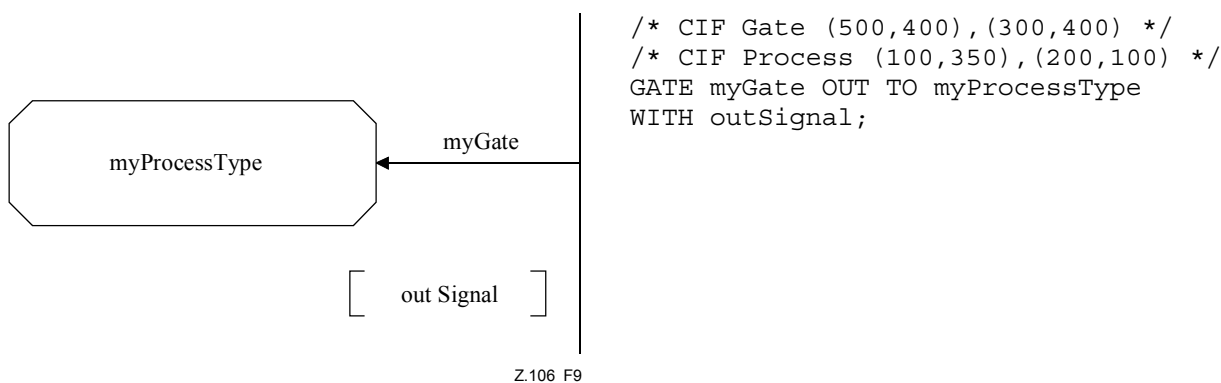


Figure 9/Z.106

7.4.24 A24 Référence de symbole d'accès < gate symbol reference: A24>

```

/* CIF GateSymbolReference <name> [In [ <charstring> ] ] [Out [ <charstring> ] ] <pointlist:
B18> */
<text position: B21>
[ <in signal list position: B39> ]
[ <out signal list position: B40> ]

```

Informations complémentaires:

Ce commentaire CIF est utilisé pour spécifier les accès sur les références d'agent et les références de type d'agent. Il convient de noter que ces accès n'ont pas de SDL-PR correspondant.

<text position: B21> définit la position du <name>.

Si **In** est spécifié, il y a alors une flèche sur l'accès orientée vers le symbole de référence. La position <in signal list position> définit la position de la chaîne <charstring> (qui décrit le liste des signaux) associée à **In**.

Si **Out** est spécifié, il y a alors une flèche sur l'accès qui part du symbole de référence. La position <out signal list position> définit la position de la chaîne <charstring> (qui décrit la liste des signaux) associée à **Out**.

Exemple:

```

/* CIF GateSymbolReference g In 's1, s2' Out 's3, s4' ((800,550), (900,
550)) */
/* CIF TextPosition (850,600) */
/* CIF TextPosition (750,500) In */
/* CIF TextPosition (850,600) Out */

```

7.4.25 A25 Connexion <connect: A25>

```
/* CIF Connect [ <position: B22> ]*/  
[ <text position: B21> ]  
<channel to channel connection>
```

La production pour <channel to channel connection> est définie au § 5.6.2.

Informations complémentaires:

<position: B22> est la position du canal sur le symbole de cadre.

<text position: B21> est la position du texte pour les identificateurs <external channel identifiers>, à savoir le texte en dehors du symbole de cadre.

Exemple:

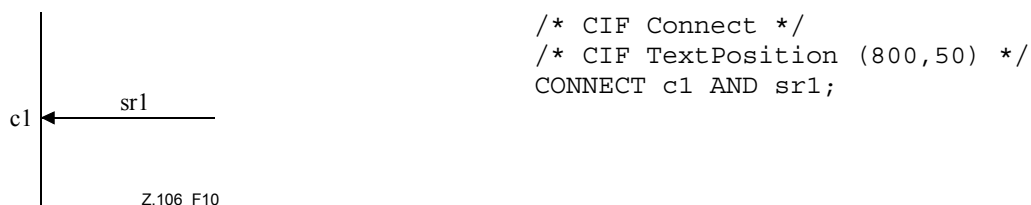


Figure 10/Z.106

7.4.26 A26 Point de connexion d'état <state connection point: A26>

```
/* CIF StateConnectionPoint <pointlist: B18> */  
[ <text position: B21> ]  
<state connection points>
```

Informations complémentaires:

il doit y avoir deux points dans la liste <pointlist: B18>. Le premier point de la liste de points doit se trouver sur le rectangle englobant du symbole de cadre du diagramme. Le second point définit l'autre extrémité du symbole de point de connexion d'état. Si une position <text position> existe, elle spécifie la position du texte dans le point <state connection point>.

7.4.27 A27 Connexion d'état <state connection: A27>

```
/* CIF StateConnection <pointlist: B18> */  
[ <text position: B21> ]  
<state partition connection>
```

La production pour <state partition connection> est définie au § 5.7.10.2.

Informations complémentaires:

le premier point de la liste de points se trouve sur le rectangle englobant du symbole correspondant au <inner entry point>. Le dernier point de la liste de points se trouve sur le rectangle englobant sur symbole de cadre de la page courante.

<text position> spécifie la position du texte dans <inner entry point>, à savoir le ou les noms du ou des points d'entrée associés au symbole d'état connecté.

7.4.28 A28 Extension de texte <text extension: A28>

```
/* CIF TextExtension <position and size: B20>  
[ { Left | Right } ] */  
[ <text position: B21> ]
```

```
[ <line: B16> ]
<text>
/* CIF End TextExtension */
```

Informations complémentaires:

Left signifie que le côté gauche du symbole est ouvert. **Right** signifie que le côté droit du symbole est ouvert. **Right** est la valeur par défaut.

Si **Left** est indiqué, le symbole et la position du texte définissent le coin supérieur *droit*.

La ligne <line: B16> est la ligne qui relie le symbole d'extension de texte à l'autre symbole. Si la ligne n'est pas spécifiée, elle sera autotracée. Le premier point de la liste de points se trouve sur le rectangle englobant du symbole d'extension de texte. Le dernier point se trouve sur le rectangle englobant le symbole auquel est associé le symbole d'extension de texte.

Un caractère nouvelle ligne avant ou après l'un des deux commentaires d'extension de texte CIF ne doit pas être considéré comme faisant partie du texte dans les symboles.

<Text extension: A28> et <Comment: A29> doivent être placés avant <end> comme suit: les extensions de texte et les commentaires peuvent être associés à une règle quelconque dans la plage <block symbol: A33> – <select symbol: A69>.

Lorsqu'une extension de texte est associée au symbole de tâche contenant une déclaration <compound statement>, l'extension de texte sera placée avant le <right curly bracket> dans la déclaration composée.

Exemple 1 (exemple avec symbole de tâche informel):

```
/* 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:

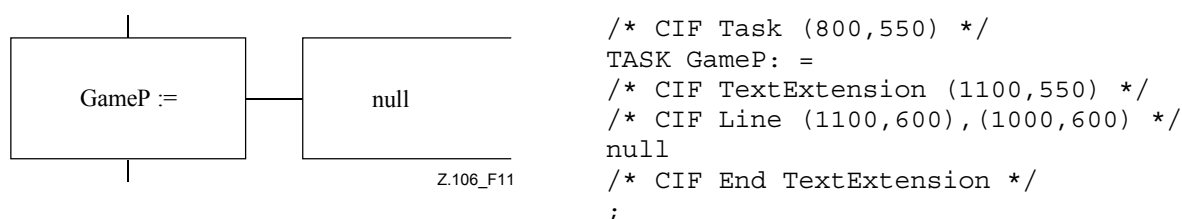


Figure 11/Z.106

7.4.29 A29 Commentaire <comment: A29>

```
/* CIF Comment <position and size: B20> [ Left | Right ] [ Dashed ] */
[ <text position: B21> ]
[ <dashed line: B17> ]
<comment> <end>
```

Informations complémentaires:

Left signifie que le côté gauche du symbole est ouvert. **Right** signifie que le côté droit du symbole est ouvert. **Right** est la valeur par défaut.

Si **Left** est indiqué, la position du symbole (et du texte) définit le coin supérieur *droit*.

Si **Dashed** est indiqué, le symbole de commentaire doit être tracé en pointillé (comme un symbole <comment symbol2> en SDL96). Si **Dashed** n'est pas indiqué, le symbole de commentaire ne doit pas être tracé en pointillé (comme un symbole <comment symbol2> en SDL92).

La ligne <dashed line: B17> est la ligne qui relie le symbole de commentaire à l'autre symbole. Si la ligne n'est pas indiquée, elle sera autotracée. Le premier point de la liste de points se trouve sur le rectangle englobant du symbole commentaire. Le dernier point de la liste de points se trouve sur le rectangle englobant du symbole auquel il est associé.

La façon d'utiliser cette construction CIF est expliquée à <text extension: A28>.

Exemple 1:

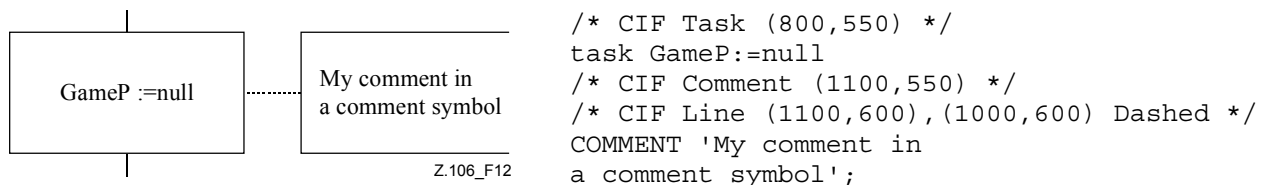


Figure 12/Z.106

Exemple associé:

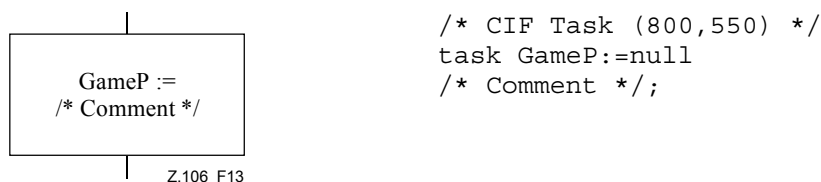


Figure 13/Z.106

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';

```

7.4.30 A30 Ligne de création <create line: A30>

/* CIF CreateLine <pointlist: B18> */

Informations complémentaires:

le premier point de la liste de points est située sur le rectangle englobant du symbole processus qui crée l'autre processus. Le dernier point se trouve sur le rectangle englobant du symbole de processus qui est créé.

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) */

```


7.4.31 A31 Ligne de flux <flow line: A31>

<line: B16>

Informations complémentaires:

ce commentaire CIF est facultatif. Si le commentaire CIF n'est pas donné pour une ligne de flux, la ligne de flux est autotracée.

Le premier point de la liste de points se trouve sur le rectangle englobant du symbole d'où provient le flux. Le dernier point se trouve sur le rectangle englobant le symbole vers lequel le flux se dirige.

Ce commentaire peut être placé avant ou après un symbole ou une construction de ligne SDL-CIF et SDL-PR dans le corps <process body> du diagramme.

Une ligne de flux joignant une autre ligne de flux doit décrire la liste de points complète depuis un point situé sur le rectangle englobant du symbole "from" jusqu'en un point situé sur le rectangle englobant du symbole "to".

Les flèches sur les lignes de flux sont implicites. Les lignes de flux après les symboles d'option de décision ou de transition doivent utiliser <answer flow line: A32>, règle CIF qui n'est pas facultative.

Exemple:

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

7.4.32 A32 Ligne de flux réponse <answer flow line: A32>

```
/* CIF Answer [ { Right | Left } ] [ InvisibleBrackets ] */  
[ <line: B16> ] [ <text position: B21> ]  
{ <textual answer part> | <textual else part> }
```

Cette règle est utilisée par <CIF descriptor: A19>.

Les productions pour <textual answer part> et <textual else part> sont définies au § 5.7.12.2.

Informations complémentaires:

ce commentaire CIF doit être utilisé pour les lignes de flux après un symbole d'option de décision ou de transition.

Si <line: B16> est indiqué:

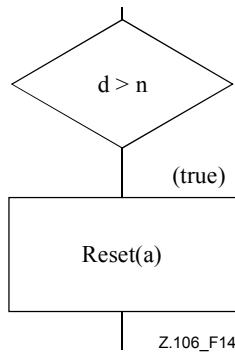
Right et **Left** n'ont pas de signification. La liste de points dans la ligne de flux spécifie si la ligne de flux commence au symbole de décision. Les règles sont les mêmes que pour <flow line: A31>.

Si <line: B16> n'est pas indiqué:

Right indique que la ligne de flux commence à la droite du symbole de décision (ou au coin inférieur droit du symbole d'option de transition). **Left** indique que la ligne de flux commence à la gauche du symbole de décision (ou au coin inférieur gauche du symbole d'option de transition). En l'absence d'indication, la ligne de flux commence en dessous du symbole de décision (ou au centre du bord inférieur du symbole d'option de transition). Le reste de la ligne de flux est autotracée.

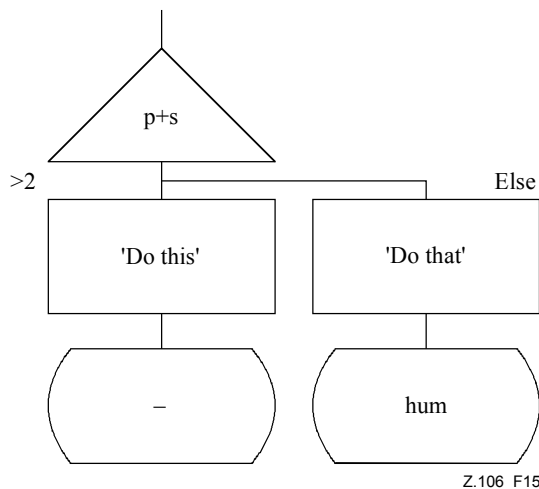
En présence de l'indication **InvisibleBrackets**, les caractères (and) qui ne peuvent pas être trouvés en SDL-PR ne sont pas visibles en SDL-GR.

Exemples:



```
/* CIF Decision (800,550) */
DECISION d > n;
/* CIF Answer */
/* CIF Line (900,650), (900,750) */
/* CIF TextPosition (910,690) */
(true):
/* CIF Reset (800,750) */
reset(a);
-----
```

Figure 14/Z.106



```
/* 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
```

Figure 15/Z.106

7.4.33 A33 Symbole de bloc <block symbol: A33>

<block symbol rectangle: B12>

[<text position: B21>]

{ <gate reference: B15>* <block reference> |

{ <gate reference: B15>* <textual typebased block definition> } }

Cette règle est utilisée par <CIF descriptor: A19>.

Informations complémentaires:

les commentaires <gate reference: B15> sont optionnels et uniquement utilisés pour spécifier les positions pour les références d'accès associées à ce symbole de bloc. Lorsque ce commentaire est omis, le texte pour cette référence d'accès sera autopositionné. Le nom de cette référence de texte dans le SDL-PR est mentionné en rapport avec le SDL-PR pour des canaux connectés.

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;
```

7.4.34 A34 Symbole de bloc en pointillé <dashed block symbol: A34>

```
/* CIF Block <position and size: B20> Dashed <block name> */  
[ <text position: B21> ]  
<gate reference: B15>*
```

Informations complémentaires:

ce commentaire CIF peut être placé partout où une référence <textual block reference> est permise. Il doit y avoir un symbole <dashed block symbol: A34> par définition <existing typebased block definition> dans le SDL-GR. La règle <gate reference: B15> est expliquée dans la règle <block symbol: A33>.

Exemple:

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

7.4.35 A35 Symbole de processus <process symbol: A35>

```
<process symbol rectangle: B13>  
[ <text position: B21> ]  
{ <gate reference: B15>* <process reference> |  
  { <gate reference: B15>* <textual typebased process definition> } }
```

Informations complémentaires:

La règle <gate reference: B15> est expliquée dans la règle <block symbol: A33>.

Exemple 1:

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

Exemple 2:

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

7.4.36 A36 Symbole de processus en pointillé <dashed process symbol: A36>

```
/* CIF Process <position and size: B20> Dashed <process name> */  
[ <text position: B21> ]  
<gate reference: B15>*
```

Informations complémentaires:

ce commentaire CIF peut être placé partout où une référence <textual process reference> est permise. Il doit y avoir un symbole <dashed process symbol: A36> par définition <existing typebased process definition> dans le SDL-GR. La règle <gate reference: B15> est expliquée dans la règle <block symbol: A33>.

Exemple:

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

7.4.37 A37 Référence d'agent dans la zone de spécification <agent reference in specification area: A37>

```
/* CIF Agent <name> [ : <type expression> ] <position and size: B20> { System | Block | Process  
} */  
[ <text position: B21> ]
```

Informations complémentaires:

ce symbole ne peut être utilisé que dans les zones de spécification.

Si **System** est spécifié, le symbole réel doit être une zone <system reference area>.

Si **Block** est spécifié, le symbole réel doit être une zone <block reference area>.

Si **Process** est spécifié, le symbole réel doit être une zone <process reference area>.

La position <text position> se réfère au texte correspondant à '<name> [: <type expression>]'.

Exemple 1:

```
/* CIF Agent P (800,550) Process */  
/* CIF TextPosition (810,560) */
```

7.4.38 A38 Symbole de paquetage <package symbol: A38>

```
/* CIF Package <position and size: B20> */  
[ <text position: B21> ]  
<package reference>
```

Informations complémentaires:

ce commentaire CIF doit seulement être utilisé pour des références de paquetage dans les diagrammes de paquetages où il existe une syntaxe SDL-PR pour les références de paquetages. Lorsqu'on décrit des références de paquetages dans des zones de spécification, la règle à utiliser est 'package ref in specification area'.

Exemple:

```
/* CIF Package (800,550) */  
PACKAGE myPKG REFERENCED;
```

7.4.39 A39 Référence de paquetage dans la zone de spécification <package reference in specification area: A39>

```
/* CIF PackageReference <name> <position and size: B20> */  
[ <text position: B21> ]
```

7.4.40 A40 Symbole d'opérateur <operator symbol: A40>

```
/* CIF Operator <identifiant> <position and size: B20> */  
[ <text position: B21> ]
```

Informations complémentaires:

ce commentaire peut être placé partout où un <entity in package>, <entity in system>, <entity in block>, <entity in process> or an <entity in procedure> peut être placé. Le symbole d'opérateur n'a pas de lien graphique direct avec la référence <textual operator reference> (qui est présentée sous forme de texte dans un symbole texte).

Exemple:

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

7.4.41 A41 Symbole de début <start symbol: A41>

```
/* CIF Start <position and size: B20> */  
[ <text position: B21> ]  
start [ <virtuality> ] [ <state entry point name> ] <end>
```

Informations complémentaires:

ce commentaire CIF doit être utilisé dans des diagrammes d'agent ou de type d'agent. Les diagrammes de procédure et d'opérateur doivent utiliser le <procedure start symbol: A55>.

Exemple:

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

7.4.42 A42 Symbole d'arrêt <stop symbol: A42>

```
/* CIF Stop <position and size: B20> */  
<stop> <end>
```

Exemple:

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

7.4.43 A43 Symbole d'état <state symbol: A43>

```
/* CIF State <position and size: B20> */  
[ <text position: B21> ]  
state <state list> <end> | state <composite state list> <end>
```

On pourra trouver des exemples pour ce commentaire CIF dans le commentaire CIF associé <nextstate symbol: A44>.

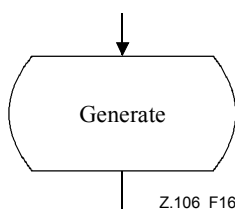
7.4.44 A44 Symbole d'état suivant <nextstate symbol: A44>

```
/* CIF NextState <position and size: B20> */  
[ <text position: B21> ]  
<nextstate>
```

Informations complémentaires:

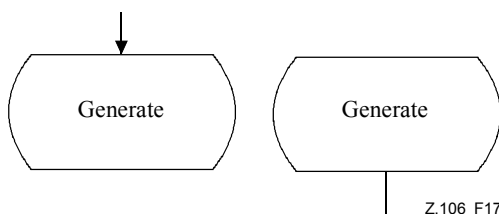
un commentaire CIF doit être fourni pour chaque état et chaque état suivant dans le SDL-PR. Cela signifie qu'il y aura deux commentaires CIF pour un symbole SDL-GR qui correspondent à un état SDL-PR et un état suivant SDL-PR. Un outil qui permet de lire un fichier CIF devra déterminer si un état et un état suivant sont en fait un symbole SDL-GR en comparant les coordonnées des symboles dans les deux commentaires CIF.

Exemples:



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

Figure 16/Z.106



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

Figure 17/Z.106

7.4.45 A45 Symbole d'entité de traitement des exceptions <exception handler symbol: A45>

```
/* CIF Exception Handler { <position and size: B20> } */  
[ <text position: B21> ]  
exceptionhandler <exception handler list> <end>
```

Exemple:

```
/* CIF Exception Handler (800,550) */
exceptionhandler myXX;
```

7.4.46 A46 Symbole sur exception active <on exception symbol: A46>

```
/* CIF OnException <position and size: B20> */
[ <text position: B21> ]
[ <pointlist> ]
onexception <exception handler name> <end>
```

Informations complémentaires:

la liste <pointlist> définit le symbole <solid on exception association symbol> qui relie le symbole <exception handler symbol> à l'autre symbole. Le premier point de la liste de points est situé sur le rectangle englobant du symbole <exception handler symbol>. Le dernier point de cette liste est situé sur le rectangle englobant du symbole auquel est associée l'exception.

Il convient de noter que le même symbole graphique <exception handler symbol> peut être utilisé comme définition d'entité de traitement des exceptions et comme sur exception. Il en est ainsi lorsque les <position and size> sont les mêmes dans les commentaires CIF Exception Handler et OnException.

Si la liste <pointlist> est omise on devra utiliser un tracé automatique pour le symbole <solid on exception association symbol>.

Exemple:

```
/* CIF OnException (800,550) */
onexception myXX;
```

7.4.47 A47 Symbole traitement <handle symbol: A47>

```
/* CIF Handle <position and size: B20> [ { Left | Right } ] */
[ <text position: B21> ]
handle [ <virtuality> ] <exception stimulus list> <end>
```

Informations complémentaires:

Left indique que la partie du symbole qui visualise une flèche est orientée vers la gauche. **Right** indique que la partie du symbole qui visualise une flèche est orientée vers la droite. La valeur par défaut est **Right**.

Exemple:

```
/* CIF Handle (800,550) */
HANDLE myException;
```

7.4.48 A48 Symbole sauvegarde <save symbol: A48>

```
/* CIF Save <position and size: B20> */
[ <text position: B21> ]
<save part>
```

Exemple:

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

7.4.49 A49 Symbole tâche <task symbol: A49>

```
/* CIF Task <position and size: B20> */
[ <text position: B21> ]
<task> <end>
```

Informations complémentaires:

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 SDL-GR contenant des armements <set> doivent utiliser la règle <set symbol: A50>; les symboles de tâches SDL-GR contenant des réinitialisations <reset> doivent utiliser la règle <reset symbol: A51>; les symboles de tâches SDL-GR contenant des exportations <export> doivent utiliser la règle <export symbol: A52>.

Exemple:

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

7.4.50 A50 Symbole d'armement <set symbol: A50>

```
/* CIF Set <position and size: B20> */  
[ <text position: B21> ]  
<set> <end>
```

Informations complémentaires:

un symbole d'armement est un symbole de tâche SDL-GR contenant une seule règle <set statement> (voir le § 11.14 de [1]).

Exemple:

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

7.4.51 A51 Symbole de réinitialisation <reset symbol: A51>

```
/* CIF Reset <position and size: B20> */  
[ <text position: B21> ]  
<reset> <end>
```

Informations complémentaires:

un symbole de réinitialisation est un symbole de tâche SDL-GR contenant une seule règle <reset statement> (voir le § 11.14 de [1]).

Exemple:

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

7.4.52 A52 Symbole d'exportation <export symbol: A52>

```
/* CIF Export <position and size: B20> */  
[ <text position: B21> ]  
<export> <end>
```

Informations complémentaires:

un symbole d'exportation est un symbole de tâche SDL-GR contenant une seule règle <export statement> (voir § 11.14 de [1]).

Exemple:

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

7.4.53 A53 Symbole de demande de création <create symbol: A53>

```
/* CIF Create <position and size: B20> */  
[ <text position: B21> ]  
<create request> <end>
```

Exemple:

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

7.4.54 A54 Symbole d'appel de procédure <procedure call symbol: A54>

```
/* CIF ProcedureCall <position and size: B20> */
[ <text position: B21> ]
<procedure call> <end>
```

Exemple:

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

7.4.55 A55 Symbole de début de procédure <procedure start symbol: A55>

```
/* CIF ProcedureStart <position and size: B20> */
[ <text position: B21> ]
start [ <virtuality> ] [ <state entry point name> ] <end>
```

Informations complémentaires:

ce commentaire CIF doit être utilisé pour les diagrammes de procédures et d'opérateurs. Les diagrammes d'agent, de type d'agent, d'état et de type d'état doivent utiliser la règle <start symbol: A41>.

Exemple:

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

7.4.56 A56 Symbole de retour <return symbol: A56>

```
/* CIF Return <position and size: B20> */
[ <text position: B21> ]
<return> <end>
```

Exemple:

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

7.4.57 A57 Symbole déclenchement d'exception <raise symbol: A57>

```
/* CIF Raise <position and size: B20> */
[ <text position: B21> ]
<raise> <end>
```

Exemple:

```
/* CIF Raise (800,550) */
RAISE myException;
```

7.4.58 A58 Symbole de décision <decision symbol: A58>

```
/* CIF Decision <position and size: B20> */
[ <text position: B21> ]
DECISION <question> <end>
```

Informations complémentaires:

une ligne de flux partant d'un symbole de décision doit être décrite par la règle <answer flow line: A32>.

Exemple: (sans lignes de flux)

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

7.4.59 A59 Symbole de signal continu <continuous signal symbol: A59>

```
/* CIF ContinuousSignal <position and size: B20> */
[ <text position: B21> ]
provided [ <virtuality> ] <continuous expression> <end> [ priority <priority name> <end> ]
```

Exemple:

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

7.4.60 A60 Symbole de condition de validation <enabling condition symbol: A60>

```
/* CIF EnablingCondition <position and size: B20> */
[ <text position: B21> ]
<enabling condition>
```

Exemple:

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

7.4.61 A61 Symbole d'option de transition <transition option symbol: A61>

```
/* CIF TransitionOption <position and size: B20> */
[ <text position: B21> ]
ALTERNATIVE <alternative question> <end>
```

Informations complémentaires:

une ligne de flux partant d'un symbole d'option de transition doit être décrite par la règle <answer flow line: A32>.

Exemple:

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

7.4.62 A62 Symbole de branchement <join symbol: A62>

```
/* CIF Join { <position and size: B20> | Invisible } */
[ <text position: B21> ]
<join> <end>
```

Informations complémentaires:

le mot clé **Invisible** signifie que le branchement SDL-PR ne doit pas être tracé comme un symbole dans le SDL-GR. Ce type de branchement en SDL-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;
```

7.4.63 A63 Connexion <Connect: A63>

```
/* CIF Connect */  
[ <line: B16> ] [ <text position: B21> ]  
connect [<virtuality>] [<connect list>] <end>
```

Informations complémentaires:

ce commentaire CIF doit être utilisé pour des lignes de flux allant d'un symbole d'état à une transition sans signal de début d'entrée ou sans signal continu.

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

```
/* CIF Label { <position and size: B20> | Invisible } */  
[ <text position: B21> ]  
{ <label> | CONNECTION <label> }
```

Informations complémentaires:

le mot clé **Invisible** signifie que l'étiquette SDL-PR ne doit pas être tracée comme un symbole dans le SDL-GR. Ce type d'étiquette en SDL-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: A62>.

Le commentaire CIF correspondant à la première étiquette <label> dans une action < free action> est placé avant CONNECTION; 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:
```

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

```
/* CIF Input <position and size: B20> [ { Left | Right } ] */  
[ <text position: B21> ]  
{ input [<virtuality>] <input list> <end> | input [<virtuality>] <spontaneous designator> <end> }
```

Informations complémentaires:

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;
```

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

```
/* CIF PriorityInput <position and size: B20>  
[ { Left | Right } ] */  
[ <text position: B21> ]  
priority input [ <virtuality> ] <priority input list> <end>
```

Informations complémentaires:

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;
```

7.4.67 A67 Symbole de sortie <output symbol: A67>

```
/* CIF Output <position and size: B20> [ { Left | Right } ] */  
[ <text position: B21> ]  
<output> <end>
```

Informations complémentaires:

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;
```

7.4.68 A68 Symbole de texte <text symbol: A68>

```
/* CIF Text <position and size: B20> */  
[ <text position: B21> ]  
<text>  
/* CIF End Text */
```

Informations complémentaires:

<text> peut ne pas contenir de correspondance de ligne '/* CIF End Text */'

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 dans le symbole de texte.

Exemple:

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

7.4.69 A69 Symbole de sélection <select symbol: A69>

```
/* CIF Select <pointlist: B18> */  
[ <text position: B21> ]  
SELECT IF <boolean simple expression> <end>
```

Informations complémentaires:

les points constituant la liste de points 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);
```

7.4.70 A70 Fin de descripteur <descriptor end: A70>

```
{ /* CIF End Decision */ ENDDECISION <end> |  
/* CIF End State */ ENDSTATE [ <state name> ] <end> |  
/* CIF End Label */ ENDCONNECTION [ <connector name> ] <end> |
```

```

/* CIF End Select */ ENDSELECT <end> |
/* CIF End TransitionOption */ ENDALTERNATIVE <end> |
/* CIF End ExceptionHandler */ ENDEXCEPTIONHANDLER [ <exception handler name> ]
<end> }

```

Informations complémentaires:

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.

7.4.71 A71 Référence de type <type reference: A71>

```

/* CIF TypeReference <position and size: B20> [ Iconized ] */
[ <id text position: B23> ] [ <stereotype text position: B24> ] [ <icon symbol: B29> ]
[ <class symbol first line position: B27> ] [ <attribute text position: B25> ]
[ <class symbol second line position: B28> ] [ <behaviour text position: B26> ]
{ {system | block | process | object | value | state} type } | signal | procedure | interface}
<identifiant> <type reference properties>

```

Informations complémentaires:

si **Iconized** fait partie du commentaire CIF, le symbole type de référence ne doit pas utiliser le symbole de classe mais les symboles spéciaux pour les types ou procédures de bloc/processus/état. Dans les autres cas, il faut utiliser le symbole de classe.

Si le symbole <icon symbol: B29> est présent, le symbole de classe doit utiliser la petite icône correspondant à la catégorie de type. Si le symbole <icon symbol: B29> est absent, il faut utiliser le nom de stéréotype.

Les symboles de classe sont répartis en trois compartiments utilisant deux lignes. Le compartiment supérieur du symbole de classe contient l' <identifiant> et le stéréotype (un **system**, **block**, **process**, **object**, **value**, **state**, **signal**, **procedure**, **interface**), ou une petite icône représentant le stéréotype. Le compartiment intermédiaire contient la partie <attribute property>. Le compartiment inférieur contient la partie <behaviour property>.

Toutes les positions de texte et de ligne sont facultatives dans cette directive.

Exemple:

```

/*CIF TypeReference (500,400) */
/* CIF Icon */
/* CIF Line1 (500,440) */
/* CIF Line2 (500,480) */
process type P
with a integer;
with signal s ( Integer );
referenced;

```

7.4.72 A72 Association <association: A72>

```

/* CIF Association <pointlist: B18> */
[<association name text position: B30>]
[<first association end role name position: B31>]
[<first association end visibility position: B32>]
[<first association end multiplicity position: B33>]
[<first association end ordering position: B34>]
[<second association end role name position: B35>]
[<second association end visibility position: B36>]
[<second association end multiplicity position: B37>]
[<second association end ordering position: B38>]
<association definition>

```

Informations complémentaires:

toutes les positions (à l'exception de <pointlist: B18>) sont facultatives.

Le premier point et le dernier point de la liste de points doivent être situés sur le rectangle englobant les symboles liés.

Exemple:

```
/* CIF Association (400,300), (500,700) */  
/* TextPosition (500,500) Association */  
association ExampleAssociation from Pt1 from Pt2;
```

7.4.73 A73 Ligne de spécialisation <specialization line: A73>

```
/* CIF SpecializationLine <pointlist: B18> */
```

Informations complémentaires:

le premier point de la liste de points est situé sur le rectangle englobant le supertype. Le dernier point est situé sur le rectangle englobant le sous-type.

7.4.74 A74 Ligne de dépendance <dependency line: A74>

```
/* CIF DependencyLine <pointlist: B18> */
```

Informations complémentaires:

le premier point de la liste de points est situé sur le rectangle englobant le client (l'entité qui en "fait usage"). Le dernier point est situé sur le rectangle englobant le fournisseur (le paquetage qui "est utilisé").

7.5 Syntaxe CIF-GR – Règles CIF de type B

Les règles CIF de type B sont des règles utilitaires auxquelles renvoient les règles de type A et ne correspondent pas à des symboles autonomes.

7.5.1 B1 Parties de diagramme <diagram parts: B1>

```
{ <page declaration: B2>+ | <nested frame: B10> }
```

Informations complémentaires:

si le diagramme est graphiquement inclus dans le diagramme englobant, un commentaire <nested frame: B10> doit être utilisé.

Dans le cas contraire, une déclaration <page declaration: B2> doit être spécifiée pour chaque page du diagramme.

7.5.2 B2 Déclaration de page <page declaration: B2>

```
/* CIF Page <page name> <size point: B22> */  
[ <frame declaration: B11> ]  
[ <diagram heading text position: B21> ]  
[ <page text position: B19> ]  
[ <package use symbol: B14> ]
```

Informations complémentaires:

si la déclaration <frame declaration: B11> 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: B2>, soit un changement <page switch: A21>, n'est pas rencontré.

La position <page text position: B19> définit le coin supérieur *droit* du rectangle englobant le texte.

La référence <package reference: B14> ne peut être spécifiée que si le symbole d'utilisation de paquetage est visible dans le SDL-GR.

Exemple:

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

7.5.3 B3 Symbole de contrainte d'accès <gate constraint symbol: B3>

<block symbol rectangle: B12> | <process symbol rectangle: B13>

7.5.4 B4 Position de texte pour la première liste de signaux <first signallist text position: B4>

```
/* CIF TextPosition <point: B22> SignalList1 */
```

Informations complémentaires:

cette position de texte correspond à la première liste de signaux mentionnée dans la spécification SDL-PR.

Exemple:

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

7.5.5 B5 Position de texte pour la seconde liste de signaux <second signallist text position: B5>

```
/* CIF TextPosition <point: B22> SignalList2 */
```

Informations complémentaires:

cette position de texte correspond à la seconde liste de signaux mentionnée dans la spécification SDL-PR.

7.5.6 B6 Position de la première flèche <first arrow position: B6>

```
/* CIF Arrow1Position <point: B22> */
```

Informations complémentaires:

ce commentaire définit la position de la flèche pour la direction mentionnée en premier dans le SDL-PR pour le canal <channel: A22>.

Exemple:

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

7.5.7 B7 Position de la seconde flèche <second arrow position: B7>

```
/* CIF Arrow2Position <point: B22> */
```

Informations complémentaires:

ce commentaire définit la position de la flèche pour la direction mentionnée en second dans le SDL-PR pour le canal <channel: A22>.

7.5.8 B8 Texte d'accès entrant <inlet text: B8>

```
/* CIF InletText <macro label> */  
[ <text position: B21> ]
```

7.5.9 B9 Texte d'accès sortant <outlet text: B9>

```
/* CIF OutletText <macro label> */  
[ <text position: B21> ]
```

7.5.10 B10 Cadre imbriqué <nested frame: B10>

```
/* CIF NestedFrame <position and size: B20> */  
[ <diagram heading text position: B21> ]
```

7.5.11 B11 Déclaration de cadre <frame declaration: B11>

```
/* CIF Frame <position and size: B20> */
```

7.5.12 B12 Rectangle de symbole de bloc <block symbol rectangle: B12>

```
/* CIF Block <position and size: B20> */
```

7.5.13 B13 Rectangle de symbole de processus <process symbol rectangle: B13>

```
/* CIF Process <position and size: B20> */
```

7.5.14 B14 Symbole d'utilisation de paquetage <package use symbol: B14>

```
/* CIF Use <position and size: B20> */  
[<text position: B21>]
```

Informations complémentaires:

ce commentaire CIF comment est utilisé pour spécifier la position du symbole d'utilisation de paquetage symbol. <text position> spécifie la position du coin supérieur gauche de la case limitant pour les articles relatifs à l'utilisation du paquetage.

7.5.15 B15 Référence d'accès <gate reference: B15>

```
/* CIF GateReference [ <name> ] <connection point: B22> */  
<text position: B21>
```

Informations complémentaires:

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 SDL-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: A22>. Voir aussi les 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: A25>.

Exemple:

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

7.5.16 B16 Ligne <line: B16>

```
/* CIF Line <pointlist: B18> */
```

7.5.17 B17 Ligne en pointillé <dashed line: B17>

```
/* CIF Line <pointlist: B18> Dashed */
```

7.5.18 B18 Liste de points <pointlist: B18>

```
<point: B22> {, <point: B22> }+
```

7.5.19 Position de texte de page B19 <page text position: B19>

```
/* CIF TextPosition <point: B22> PageName */
```

7.5.20 B20 Position et taille <position and size: B20>

<position point: B22> [, <size point: B22>]

Informations complémentaires:

la taille <size point: B22> peut être omise si une taille a été définie par défaut avant au moyen du commentaire <default size: A20>. La position <position point: B22> correspond au coin supérieur *gauche* du rectangle englobant s'il n'y a rien de spécifié dans des commentaires placés avant.

7.5.21 B21 Position de texte <text position: B21>

/* CIF TextPosition <point: B22> */

Informations complémentaires:

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) */
```

7.5.22 B22 Point <point: B22>

(<integer>, <integer>)

7.5.23 B23 Position du texte de l'identificateur <id text position: B23>

/* CIF TextPosition <point: B22> TypeRefId */

Informations complémentaires:

cette position de texte définit le coin supérieur gauche du rectangle englobant le texte pour la partie identificateur d'un symbole de référence de type.

7.5.24 B24 Position du texte de stéréotype <stereotype text position: B24>

/* CIF TextPosition <point: B22> Stereotype */

Informations complémentaires:

cette position de texte définit le coin supérieur gauche du rectangle englobant le texte pour la partie stéréotype d'un symbole de référence de type.

7.5.25 B25 Position du texte d'attribut <attribute text position: B25>

/* CIF TextPosition <point: B22> Attribute */

Informations complémentaires:

cette position de texte définit le coin supérieur gauche du rectangle englobant le texte pour la partie attribut d'un symbole de référence de type.

7.5.26 B26 Position du texte de comportement <behavior text position: B26>

/* CIF TextPosition <point: B22> Behavior */

Informations complémentaires:

cette position de texte définit le coin supérieur gauche du rectangle englobant le texte pour la partie comportement d'un symbole de référence de type.

7.5.27 B27 Position de la première ligne du symbole de classe <class symbol first line position: B27>

/* CIF Line1 <point: B22> */

Informations complémentaires:

ce commentaire CIF est utilisé pour définir la position d'une ligne séparant le compartiment supérieur et le compartiment intermédiaire d'un symbole de classe. Le point définit le point de terminaison gauche de la ligne.

Exemple:

```
/* CIF Line1 (800,550) */
```

7.5.28 B28 Position de la deuxième ligne du symbole de classe <class symbol second line position: B28>

```
/* CIF Line2 <point: B22> */
```

Informations complémentaires:

ce commentaire CIF est utilisé pour définir la position d'une ligne séparant le compartiment supérieur et le compartiment intermédiaire d'un symbole de classe. Le point définit le point de terminaison gauche de la ligne.

Exemple:

```
/* CIF FirstLine (900,550) */
```

7.5.29 B29 Symbole icône <icon symbol: B29>

```
/* CIF Icon [ <position and size: B20>] */
```

7.5.30 B30 Position de texte de nom d'association <association name text position: B30>

```
/* CIF TextPosition <point: B22> Association */
```

Informations complémentaires:

cette position de texte définit le coin supérieur gauche du rectangle englobant le texte du nom d'association.

7.5.31 B31 Position du nom de rôle de la première fin d'association <first association end role name position: B31>

```
/* CIF TextPosition <point: B22> Role1 */
```

Informations complémentaires:

cette position de texte définit le coin supérieur gauche du rectangle englobant le texte du nom de rôle.

7.5.32 B32 Position de visibilité de la première fin d'association <first association end visibility position: B32>

```
/* CIF TextPosition <point: B22> Visibility1 */
```

Informations complémentaires:

cette position de texte définit le coin supérieur gauche du rectangle englobant le texte du texte de visibilité.

7.5.33 B33 Position de multiplicité de la première fin d'association <first association end multiplicity position: B33>

```
/* CIF TextPosition <point: B22> Multiplicity1 */
```

Informations complémentaires:

cette position de texte définit le coin supérieur gauche du rectangle englobant le texte de multiplicité.

7.5.34 B34 Position d'ordre de la première fin d'association <first association end ordering position: B34>

/* CIF TextPosition <point: B22> Ordering1 */

Informations complémentaires:

cette position de texte définit le coin supérieur gauche du rectangle englobant le texte d'ordre.

7.5.35 Position du nom de rôle de la deuxième fin d'association B35 <second association end role name position: B35>

/* CIF TextPosition <point: B22> Role2 */

Informations complémentaires:

cette position de texte définit le coin supérieur gauche du rectangle englobant le texte du nom de rôle.

7.5.36 B36 Position de visibilité de la deuxième fin d'association <second association end visibility position: B36>

/* CIF TextPosition <point: B22> Visibility2 */

Informations complémentaires:

cette position de texte définit le coin supérieur gauche du rectangle englobant le texte de visibilité.

7.5.37 B37 Position de multiplicité de la deuxième fin d'association <second association end multiplicity position: B37>

/* CIF TextPosition <point: B22> Multiplicity2 */

Informations complémentaires:

cette position de texte définit le coin supérieur gauche du rectangle englobant le texte de multiplicité.

7.5.38 B38 Position d'ordre de la deuxième fin d'association <second association end ordering position: B38>

/* CIF TextPosition <point: B22> Ordering2 */

Informations complémentaires:

cette position de texte définit le coin supérieur gauche du rectangle englobant le texte du nom.

7.5.39 B39 Position de la liste des signaux d'entrée <in signal list position: B39>

/* CIF TextPosition <point: B22> In */

Informations complémentaires:

cette position de texte définit le coin supérieur gauche du rectangle englobant le texte de la liste des signaux d'entrée associée à un symbole d'accès.

7.5.40 B40 Position de la liste des signaux de sortie <out signal list position: B40>

/* CIF TextPosition <point: B22> Out */

Informations complémentaires:

cette position de texte définit le coin supérieur gauche du rectangle englobant le texte de la liste des signaux de sortie associée à un symbole d'accès.

7.6 Commentaires CIF spécifiques aux outils

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

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

Informations complémentaires:

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 'The CIF comment associated with rule Ax'*/  
/* CIF Specific mySDLTool 'This information is understood by mySDLTool.  
Tools other than mySDLTool may or may not understand this information.  
This tool-specific CIF comment is associated with rule Ax'*/  
'This is the SDL-PR associated with A rule number X';
```

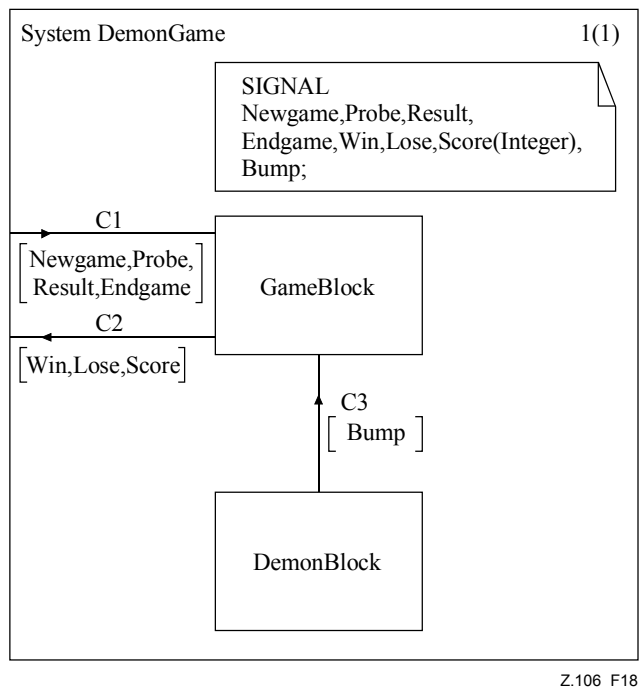
8 Exemples

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.

A noter 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.

8.1 Système du jeu DemonGame

8.1.1 Système du jeu DemonGame



```

/* 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 End Text */
/* 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;

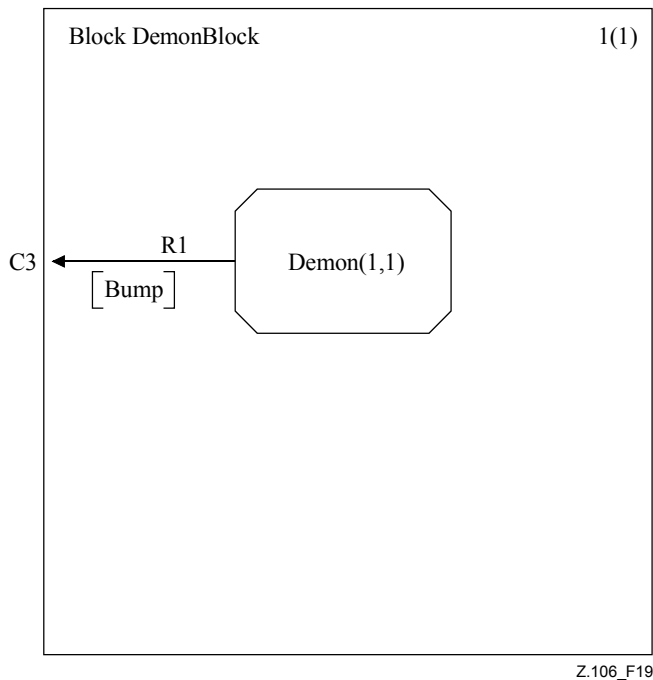
```

SDL-GR

SDL-CIF

Figure 18/Z.106 – Système du jeu DemonGame

8.1.2 Bloc DemonBlock



SDL-GR

```

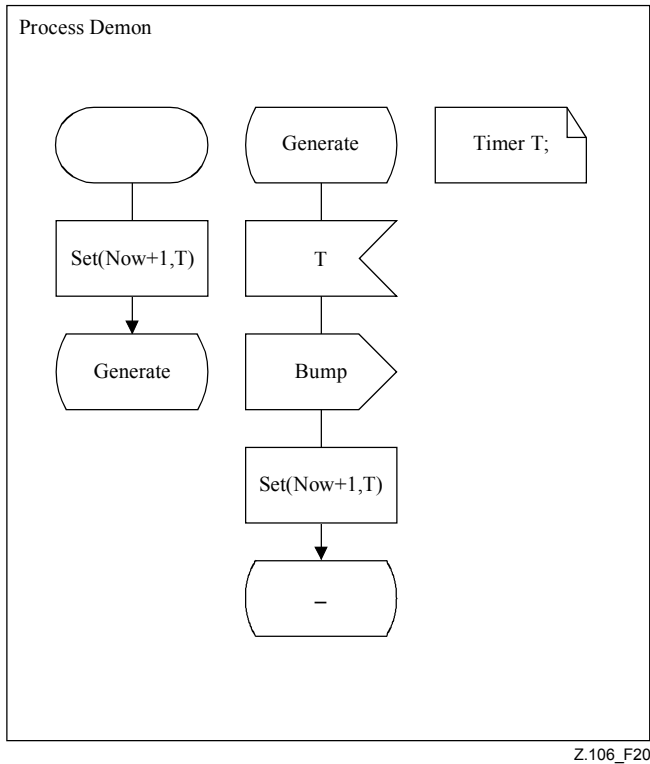
/* CIF BlockDiagram */
/* CIF Page 1 (1000,1000) */
/* CIF Frame (100,100), (800,800) */
*/
Block DemonBlock;
/* CIF Specific mySDLTool Page 1
NoGrid */
/* CIF DefaultSize (300,200) */
/* CIF Connect */
/* CIF TextPosition (25,300) */
Connect C3 and R1;
/* CIF SignalRoute
(250,350), (0,350) */
/* CIF 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;

```

SDL-CIF

Figure 19/Z.106 – Block DemonBlock

8.1.3 Processus Demon



SDL-GR

```

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

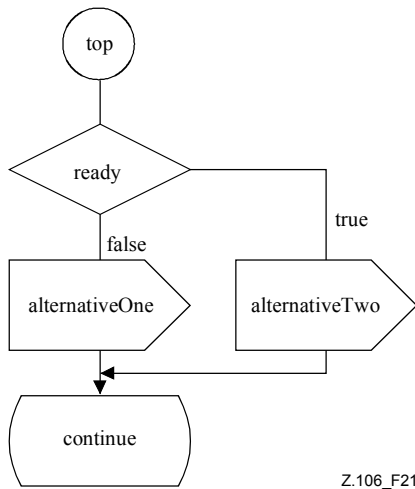
SDL-CIF

Figure 20/Z.106 – Processus Demon

8.2 Constructions SDL complexes

8.2.1 Jonction de lignes de flux, exemple 1

Noter 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;
```

Figure 21/Z.106 – Jonction de lignes de flux

8.2.2 Jonction de lignes de flux, exemple 2

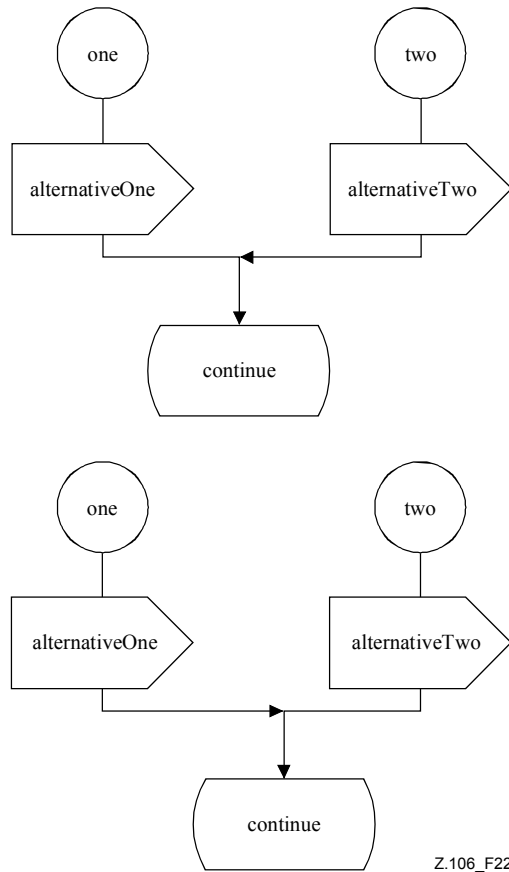
Cet exemple est très semblable au précédent. C'est en fait le même SDL-GR, mais le SDL-PR est différent. Noter que, dans le SDL-PR, les deux lignes de flux qui se rejoignent sont placées à proximité de leurs symboles de départ.

```
/* 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;
```

8.2.3 Jonction de lignes de flux, exemple 3

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

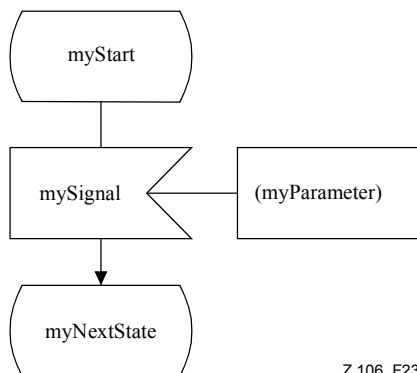


```
/* CIF Label (150,100), (100,100)
*/
connection
  one:
    /* CIF Output (100,300) */
    output alternativeOne;
    /* CIF Label Invisible */
    grs0:
      /* CIF NextState (250,500) */
      nextstate continue;
/* CIF End Label */
endconnection one;
/* CIF Label (450,100), (100,100)
*/
connection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

Figure 22/Z.106 – Jonction de lignes de flux

8.2.4 Lignes et rectangles englobants

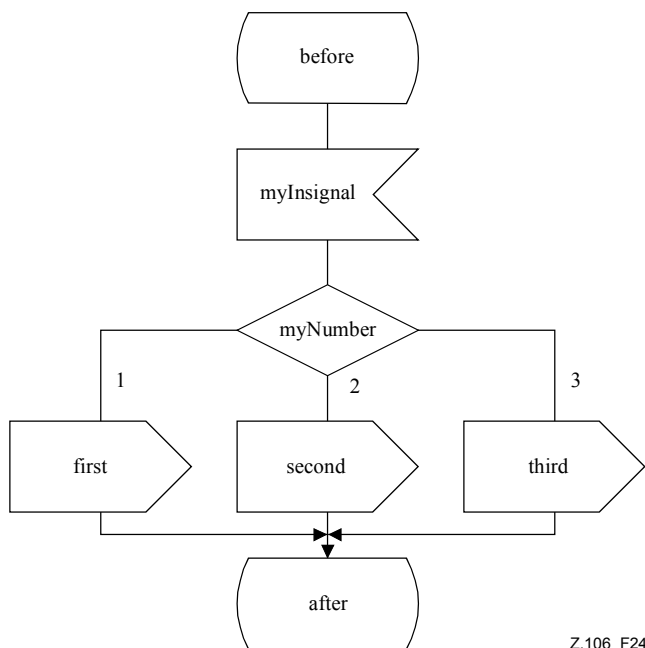
Noter que la ligne d'extension de texte est décrite comme si ses points terminaux étaient sur le bord des rectangles englobant les symboles associés, même si cela n'est pas complètement Vrai dans le SDL-GR. (Le point terminal du symbole d'entrée est à l'intérieur du rectangle englobant). Cette règle s'applique à toutes les constructions de type lignes 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;
```

Figure 23/Z.106 – Lignes et rectangles englobants

8.2.5 Lignes de flux de réponse placées après une décision



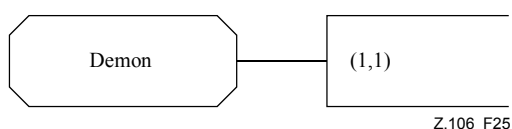
```

/* 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 Estate */
endstate;
  
```

Figure 24/Z.106 – Lignes de flux de réponse

8.2.6 Extension de texte

L'extension de texte suivante est autorisée:

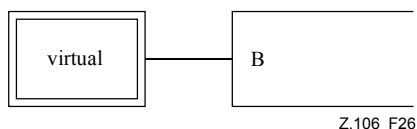


```

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

Figure 25/Z.106 – Extension de texte

Celle-ci aussi:



```

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

Figure 26/Z.106 – Extension de texte

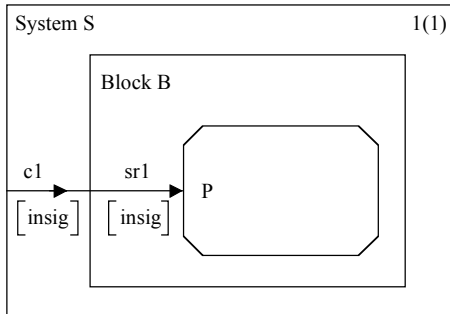
8.2.7 Diagrammes imbriqués

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 externe le canal interne est connecté;
- 4) nommer la page avec un nom approprié.

a) *Forme imbriquée*

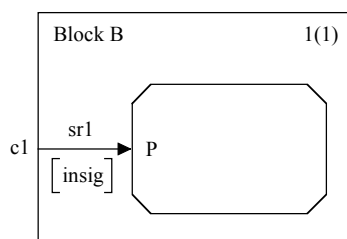
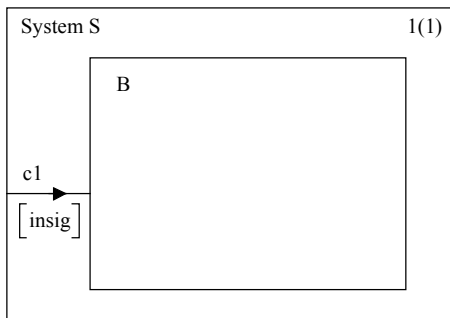


```

/* 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 Channel (100,250), (200,250) */
CHANNEL 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) *Après conversion/sans 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;

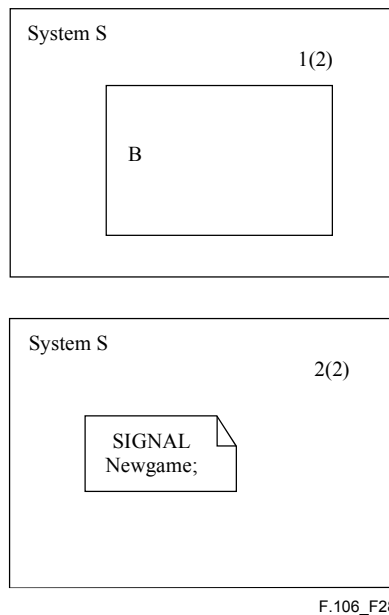
/* CIF BlockDiagram */
/* CIF Page 1 (600,500) */
/* CIF Frame (100,100), (400,300) */
Block B;
/* CIF Channel (100,250), (200,250) */
CHANNEL sr1
FROM ENV TO P WITH insig;
/* CIF Connect */
CONNECT c1 AND sr1;
/* CIF Process (200,200), (200,100) */
PROCESS P REFERENCED;
/* CIF End BlockDiagram */
ENDBLOCK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

```

Z.106_F27

Figure 27/Z.106 – Diagrammes imbriqués

8.2.8 Plusieurs pages



```

/* 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 End Text */
/* CIF End SystemDiagram */
ENDSYSTEM S;

```

Figure 28/Z.106

8.3 Situations non traitées par le CIF

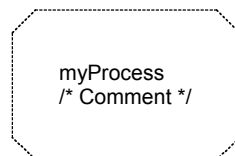


Figure 29/Z.106

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 SDL-GR est perdu. La conséquence est que le CIF ne peut pas être utilisé comme format de stockage pour tous les diagrammes SDL-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 qui apparaissent dans le CIF du fait que le SDL-PR n'équivaut pas totalement au SDL-GR. Des problèmes similaires pour introduire des commentaires textuels SDL-PR 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 SDL-PR?

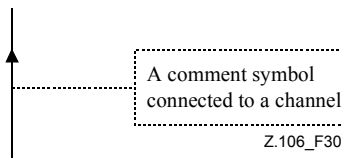


Figure 30/Z.106

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 SDL-PR. (La conversion la plus fidèle, de ce SDL-GR en SDL-PR, serait de dupliquer les symboles de commentaire et d'extension; mais que se passerait-il si la traduction en SDL-PR devait être retraduite en SDL-GR?).

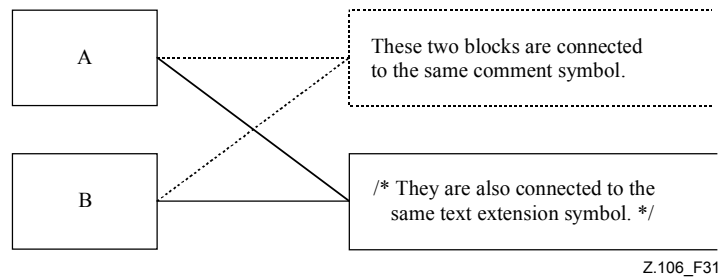


Figure 31/Z.106

Voici une situation que le CIF ne peut pas traiter, en raison des principes mêmes du CIF. Cette situation correspond à une partie de SDL-GR contenant quelque chose de ressemblant à du CIF:

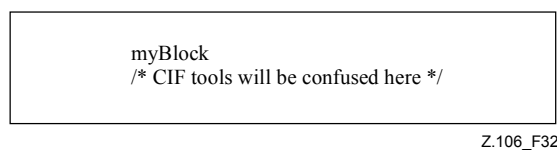


Figure 32/Z.106

9 Critères de conformité au CIF

9.1 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.

9.2 Tracé automatique/tracé forcé

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.

9.3 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 la présente 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.

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 continuellement de nouvelles directives spécifiques aux outils.

Tout d'abord, le nom des outils doit être approuvé par le Groupe de travail chargé de la Rec. UIT-T 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 au Groupe de travail chargé de la Rec. UIT-T Z.106. Ainsi, ces nouvelles directives pourront être intégrées dans cette Recommandation, en tant que directives recommandées non spécifiques aux outils, obligatoires ou facultatives.

Les réussites ou les échecs d'implémentation du SDL-CIF devront être signalés au Groupe de travail chargé de la Rec. UIT-T Z.106, afin d'ajuster la classification des directives en obligatoires ou facultatives.

Une liste actualisée des commentaires CIF spécifiques aux outils, qui sont utilisés dans les outils du marché, est accessible sur le site Web de l'UIT. En outre, le présent appendice contient une liste initiale des commentaires CIF spécifiques aux outils qui étaient connus lors de la rédaction du présent appendice.

I.2 Commentaires CIF spécifiques aux outils actuellement disponibles

Les commentaires CIF spécifiques aux outils 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 paragraphe de la présente Recommandation relatif aux commentaires CIF spécifiques.

Dans la suite du présent appendice, nous utiliserons un outil imaginaire dont le nom <tool name> est mySDLTTool. Un développeur d'outil devra substituer mySDLTTool par le nom approprié de son outil et qui identifie de manière univoque son outil. Les mots clés spécifiques aux outils définis

dans le présent appendice ne figurent pas dans l'index des mots clés de la présente Recommandation.

I.2.1 Position des commentaires CIF spécifiques aux outils

Ce point est abordé dans les informations complémentaires spécifiques associées au commentaire <tool-specific CIF comments: C0>).

I.2.2 Exemple

Dans l'exemple ci-après, la majorité des commentaires CIF spécifiques-outils est associé à la règle CIF <system diagram: A6>. Des détails sur chacun de ces commentaires sont donnés plus loin dans le présent appendice.

```
/* 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 */
...
```

I.2.3 C1 Numéro de version de l'outil <tool version number: C1>

```
/* CIF Specific mySDLTool Version x.y */
<diagram description: A2>
```

Association:

ce commentaire CIF spécifique à un outil peut être associé à une règle A mentionnée dans la règle <diagram start: A3>.

Exemple (dans lequel cette règle spécifique à l'outil est associée à la règle <system diagram start: A6>):

```
/* CIF SystemDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100), (1700,2100) */
/* CIF Specific mySDLTool Version 1.2 */
SYSTEM mySystem;
```

I.2.4 C2 Fichier d'origine <original file: C2>

```
/* CIF Specific mySDLTool OriginalFileName <file name char string literal> */
<diagram start: A3>
```

Association:

ce commentaire CIF spécifique à un outil peut être associé à une règle A mentionnée dans la règle <diagram start: A3>.

Informations complémentaires:

ce commentaire est utilisé pour mémoriser le nom du fichier d'origine qui a été converti en CIF. Quand le fichier CIF est reconverti 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 charstring literal>, et non le chemin complet (par exemple, /home/lat/a.ssy).

Exemple (dans lequel cette règle spécifique à l'outil est associée à la règle <system diagram start: A6>):

```
/* CIF SystemDiagram */
/* CIF Page 1 (1900,2300) */
/* CIF Frame (100,100),(1700,2100) */
/* CIF Specific mySDLTool OriginalFileName 'mysystem.ssy' */
SYSTEM mySystem;
```

I.2.5 C3 Spécification de détails pour la page <page details specification: C3>

/* CIF Specific mySDLTool Page <page number> [ShowMeFirst] [Scale <integer>] [Grid <point: B22>] [AutoNumbered] [FixedHeadingSize [<fixed size point: B22>]] */

Association:

ce commentaire CIF spécifique à un outil peut être associé à une règle A mentionnée dans la règle <diagram start: A3>. Il peut y avoir zéro ou un commentaire de ce type par déclaration <page declaration: B2> dans le diagramme.

Informations complémentaires:

ce commentaire spécifique à l'outil est utilisé afin de fournir des détails supplémentaires pour une page déjà définie:

- le mot clé "ShowMeFirst" indique que la page considérée doit apparaître comme page par défaut lorsque le diagramme est chargé dans un éditeur. Dans le cas contraire, ce mot clé ne doit pas être spécifié;
- 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" définit 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" et ainsi de suite. 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 de un. Un seul littéral <page name charstring literal> doit être spécifié dans une déclaration <page declaration: B2> 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.

En outre, si mot clé "FixedHeadingSize" est spécifié, ce commentaire CIF indique que le symbole de tête additionnel devra avoir une taille fixe et non adaptée à la taille du texte. 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 utilise ce commentaire pour les symboles de tête additionnels que l'utilisateur aura masqué le texte par un double clic.

Exemple (dans lequel cette règle spécifique à l'outil est associée à la règle <system diagram start: A6>):

```
/* 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 */
```

1.2.6 C4 Taille fixe <fixed size: C4>

/* CIF Specific mySDLTool FixedSize [<fixed size point: B22>] */

Association:

ce commentaire CIF spécifique à un outil est associé à la règle <text symbol: A68>.

Informations complémentaires:

ce commentaire indique que bien qu'on ait attribué une taille au symbole qui vient d'être défini (taille qui permet de rendre visible le texte complet), une taille fixe plus petite est utilisée pour visualiser le symbole; le texte contenu n'est alors plus entièrement visible. Pour revenir à la taille initiale, l'utilisateur devra cliquer deux fois à l'intérieur du symbole. L'outil SDL utilise ce commentaire pour des symboles texte que l'utilisateur a 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	Réseaux câblés et transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, circuits téléphoniques, télégraphie, télécopie et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux de données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information et protocole Internet
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication