5i’

D.3.8.2.1 *D´etermination des ´etats requis*

L’auteur d’un diagramme LDS dispose g´en´eralement d’une certaine latitude pour d´efinir les ´etats d’un processus. Il peut avoir besoin d’une strat´egie qui lui permette d’identifier les ´etats du processus et cette strat´egie peut | tre informelle ou formelle. Un jugement s | r (que seule une longue exp´erience permet d’acqu´erir) est n´ecessaire si l’on veut ´etablir des diagrammes LDS tels que l’identification d’un trop grand nombre d’´etats ne les complique pas inutilement ou qu’un nombre artificiellement r´eduit d’´etats ne les emp | che d’exploiter les avantages qu’offre le LDS. Avant que l’auteur ne commence à ´etablir le diagramme, certaines ´etapes pr´eliminaires (´etudi´ees au § D.3.2) doivent | tre achev´ees, par exemple:

— la structuration du syst`eme en blocs fonctionnels;

— la repr´esentation d’un ou plusieurs processus LDS par bloc;

— le choix des signaux d’entr´ee et de sortie;

— l’utilisation des donn´ees dans le processus.

Tous les facteurs ci-dessus exercent un effet important dans la d´etermination des ´etats de chaque processus. L’effet qu’exerce le choix des signaux d’entr´ee sur le nombre d’´etats d’un diagramme LDS est illustr´e par les deux exemples de la figure D-3.8.11.

Figure D-3.8.11, p. 1

D.3.8.2.2 *R´eduction du nombre des ´etats*

Ayant appliqu´e une strat´egie pour l'identification des ´etats d'un processus, l'auteur d'un diagramme LDS estimera peut- | tre qu'un trop grand nombre d'´etats ont ´et´e utilis´es. Le nombre des ´etats est important car la dimension et la complexit´e d'un diagramme LDS sont souvent ´etroitement li´ees à ce nombre. Il existe certes des moyens qui permettent de r´eduire le nombre des ´etats mais le fait qu'un diagramme LDS soit complexe n'est pas, en soi, une raison justifiant sa modification; en effet, la complexit´e du diagramme peut
| tre simplement due à la complexit´e inh´erente au processus d´efini. Le choix de l'ensemble d'´etats doit g´en´eralement offrir le plus de clart´e possible à la s´equence d'interactions entre le processus et son environnement. Ce n'est d'ordinaire pas en minimisant le nombre d'´etats que l'on obtient cette clart´e. Le nombre de s´equences ind´ependantes trait´ees par un processus exerce un effet multiplicateur sur le nombre d'´etats. Il est donc souhaitable que les s´equences ind´ependantes des processus s´epar´es soient trait´ees de fa,con à r´eduire le nombre d'´etats et à obtenir une plus grande clart´e.

On peut r´eduire le nombre des ´etats en effectuant la s´eparation des fonctions communes ou la fusion des ´etats ou encore en recourant au concept de service. Certaines structures de donn´ees peuvent ´egalement conduire à une r´eduction du nombre des ´etats. Pour d'autres repr´esentations, l'emploi des macros peut | tre avantageux.

D.3.8.2.3 *S´eparation des fonctions communes*

Lors de la mise en place d'un diagramme LDS, on peut constater que la d´efinition d'un aspect particulier et r´ep´etitif d'un processus n´ecessite la repr´esentation d'´etats r´ep´etitifs. La figure D-3.8.12 repr´esente une partie d'un diagramme LDS d´efinissant un processus de signalisation de ligne et illustrant la condition selon laquelle une tonalit´e de signalisation de ligne doit | tre pr´esente pendant une certaine dur´ee avant que l'on consid`ere que le signal de ligne a ´et´e d´etect´e.

Pour sp´ecifier cet aspect, il convient d'ins´erer un ´etat interm´ediaire entre les ´etats aucun _signal _de _ ligne _n'est _d´etect´e et conversation. Supposons que, dans un diagramme complet, une telle fonction commune doit | tre r´ep´et´ee à chaque point où le signal est d´etect´e. Une autre solution consiste à d´efinir un processus s´epar´e qui est responsable du contr | le de la tonalit´e de signalisation de ligne et de la d´etection des signaux sur la base du temps de reconnaissance sp´ecifi´e. L'existence de ce nouveau processus permettrait de repr´esenter le diagramme de la figure D-3.8.12 de la mani`ere indiqu´ee dans la figure D-3.8.13. (Dans un contexte donn´e, les figures peuvent
| tre rendues ´equivalentes moyennant l'introduction d'un nouveau signal signal _de _ligne _valide.)

Il convient de noter la l´eg`ere diff´erence entre le processus de la figure D-3.8.12 et les deux processus de la figure D-3.8.13.

Le processus de la figure D-3.8.12 commence à compter d`es r´eception de T1 alors que le processus de traitement des appels [processus b) de la figure D-3.8.13] commence à compter d`es r´eception de signal _de _ ligne _valide qui est engendr´e et ´emis à la r´eception de T1 par le processus a) de la figure D-3.8.13. Il s'ensuit que, dans le second cas, le comptage d´emarre apr`es T1 + temps de g´en´eration, d'´emission et de r´eception du signal. En outre, d'autres signaux peuvent avoir ´et´e mis dans la file d'attente pendant le temps n´ecessaire à la g´en´eration et à l'´emission de signal _de _ligne _valide.

Une seconde particularit´e est que cette s´eparation d'une sous-fonction ne serait pas possible si le signal que doit recevoir la sous-fonction devait | tre re,cu ´egalement par la fonction principale, un signal ´etant toujours achemin´e vers un seul processus. Si, dans l'exemple, le m | me signal S _off devait | tre re,cu dans un autre ´etat où il n'est pas besoin de le valider pour le temps T1, il n'aurait pas ´et´e possible de s´eparer la sous-fonction de validation en un autre processus.

Les solutions qui font appel à des processus distincts sont g´en´eralement utiles lorsque les signaux doivent | tre trait´es ind´ependamment des ´etats dans le processus principal. Dans ce cas, les op´erations qui pr´ec`edent et qui suivent les processus peuvent traiter les s´equences d´etaill´ees et d´echarger un processus principal de tous les d´etails. Ceci engendre souvent une modularit´e utile permettant, par exemple, d'isoler les caract´eristiques particuli`eres des syst`emes de signalisation, du processus principal plus ax´e sur le service.

Une autre mani`ere de traiter le probl`eme consiste à appliquer le concept de service, expliqu´e au § D.5.3.

Une solution diff´erente du probl`eme consisterait à employer la notation MACRO, comme indiqu´e à la figure D-3.3.1. Dans ce cas, on obtient pour le diagramme la compacit´e voulue sans modifier en rien la s´emantiquedu diagramme original. Il est en outre possible d'appeler la MACRO à partir de plusieurs états si la logique du processus l'exige.

Figure D-3.8.12, p. 2

Figure D-3.8.13, p. 3

D.3.8.2.4 *Fusion des ´etats*

Si, dans un diagramme LDS, la destination future de deux ´etats est la m | me, on peut, ind´ependamment de leur ´evolution ant´erieure, effectuer la fusion de ces deux ´etats en un seul, sans affecter la logique du diagramme.

La figure D-3.8.14 repr´esente une partie d'un diagramme LDS comportant deux ´etats dont le <<pass´e>> est diff´erent mais dont le <<futur>> est identique. Dans la figure D-3.8.15, les deux ´etats ont ´et´e combin´es en un seul. Il s'agit l`a d'un exemple relativement simple dans lequel la r´eduction de la complexit´e est peu importante, mais cette m | me technique peut | tre utilis´ee pour obtenir une plus grande simplification. La s´emantique du LDS ne pr´evoit pas une d´ecision cons´ecutive à un ´etat pour d´eterminer le <<pass´e>> du processus ant´erieurement à cet ´etat (c'est-`a-dire de d´eterminer, pour cet exemple, si A6 ou B4 a ´et´e ´emis), à moins que cette information n'ait ´et´e explicitement stock´ee avant l'entr´ee dans l'´etat. A noter, que l'attribution d'un nom aux donn´ees d'une entr´ee a pour effet de mettre la valeur en m´emoire.

Un ´etat doit repr´esenter une situation logique du processus et il n'est donc pas souhaitable d'effectuer la fusion de situations logiques diff´erentes en un seul ´etat.

Des pr´ecautions sont à prendre si un diagramme fusionn´e doit | tre modifi´e ult´erieurement. L'usager devrait rechercher si la modification envisag´ee a ou non le m | me effet sur les diverses branches initiales.

Figures D-3.8.14 et D-3.8.15, C | TE-A-C | TE, p. 4

D.3.8.3 *Entr´ees*

Le pr´esent § D.3.8.3 a pour objet d’expliquer la notion d’entr´ee ainsi que l’utilisation des entr´ees dans des diagrammes LDS ne faisant pas appel à la notion de mise en r´eserve. La notion de mise en r´eserve et l’utilisation de cette notion en m | me temps que la notion d’entr´ee font l’objet du § D.3.8.4.

D.3.8.3.1 *Consid´erations g´en´erales*

Un symbole d’entr´ee attach´e à un ´etat signifie que, si le signal dont le nom figure dans le symbole d’entr´ee arrive pendant que le processus est dans cet ´etat, il faut interpr´eter la transition qui suit le symbole d’entr´ee. Lorsqu’un signal a d´eclench´e l’interpr´etation d’une transition, ce signal n’existe plus et on dit qu’il a ´et´e absorb´e.

Un signal peut | tre accompagn´e de donn´ees associ´ees. Par exemple, un signal portant le nom <<chiffre>> sert non seulement à d´eclencherl’ex´ecution d’une transition par le processus de r´eception mais encore à v´ehiculer la valeur du chiffre (0 à 9), ces donn´ees pouvant | tre utilis´ees par le processus r´ecepteur.

En LDS/PR, l’instruction INPUT contient une liste d’identificateurs de signaux. Les valeurs contenues dans les signaux sont indiqu´ees à l’aide d’identificateurs de variables. Les identificateurs de variables doivent | tre du type indiqu´e dans la d´efinition de signal, de sorte que leur position est tr`es importante. Ces identificateurs de variables sont placés entre parenthèses et séparés par des virgules (voir la figure D-3.8.16). Si une ou plusieurs valeurs de signal sont rejetées, les variables correspondantes font défaut, ce qui est indiqué par deux ou plusieurs virgules consécutives (figure D-3.8.17).

Figure D-3.8.16 [T13.100]   (a traiter comme tableau MEP), p. 6

Figure D-3.8.17 [T14.100]   (a traiter comme tableau MEP), p. 7

En LDS/GR, une entr´ee est repr´esent´ee à l'aide d'un symbole d'entr´ee contenant la liste d'identificateurs de signaux et les identificateurs de variables correspondants pour les valeurs achemin´ees. Pour pouvoir | tre communiqu´ees au processus, ces valeurs doivent | tre d´esign´ees nomm´ement dans les symboles d'entr´ee, entre parenth`eses.

On trouvera des exemples de r´eception de valeurs des entr´ees dans les figures D-3.8.18, D-3.8.19 et D-3.8.20.

Les donn´ees auxquelles un nom est assign´e peuvent | tre utilis´ees par le processus r´ecepteur quand l'entr´ee est interpr´et´ee.

La figure D-3.8.18 montre la r´eception du signal <<d´ecroch´e>>. Le signal <<d´ecroch´e>> est accompagn´e de donn´ees associ´ees (num´ero _de _l'abonn´e) avec pour valeur 9269. Le signal peut | tre re,cu de trois mani`eres, comme indiqu´e en a) et c) de la figure.

La figure D-3.8.19 montre comment envoyer et recevoir plusieurs valeurs en un seul signal. Chaque ´el´ement doit | tre s´epar´e du suivant par une virgule. La figure D-3.8.20 c) montre comment ignorer les valeurs ind´esirables en laissant un blanc dans la liste de sortes.

A noter que, dans le signal de sortie, il est impossible d'´ecrire des expressions pour E1, E2 ou E3, alors que dans le signal d'entr´ee, il faut employer des variables pour recevoir les valeurs ´emises.

Dans le LDS, il n'est pas n´ecessaire de dessiner des symboles d'entr´ee pour repr´esenter les signaux dont l'arriv´ee n´ecessiterait une transition nulle (c'est-`a-dire une transition qui ne contient aucune action et qui ram`ene au m | me ´etat). La convention admise est la suivante: pour tout signal qui n'est pas repr´esent´e dans un symbole d'entr´ee explicite à un ´etat donn´e, il existe, dans cet ´etat, un symbole d'entr´ee implicite et une transition nulle. Conform´ement à cette convention, les deux diagrammes repr´esent´es dans la figure D-3.8.21 sont logiquement ´equivalents et peuvent | tre indistinctement utilis´es.

Figure D-3.8.18, p. 8

Figure D-3.8.19, p. 9

Figure D-3.8.20, p. 10

Figure D-3.8.21, p. 11

Lorsqu’un certain nombre d’entr´ees aboutissent à la m | me transition, tous les identificateurs de signaux qui s’y rapportent peuvent
| tre plac´es dans un m | me symbole d’entr´ee. Le LDS pr´evoit une notation abr´eg´ee pour repr´esenter une entr´ee de tous les signaux (valable pour ce processus) non explicitement mentionn´es dans cet ´etat. La figure D-3.8.22 illustre cet aspect et les diagrammes qui y sont repr´esent´es sont logiquement ´equivalents. Si tous les identificateurs de signaux sont mentionn´es, il faut les s´eparer par des virgules.

D.3.8.3.2 *M´ecanisme implicite de mise en file d’attente*

Un ou plusieurs signaux peuvent | tre en attente d’absorption lorsqu’un processus parvient à un nouvel ´etat. Cela signifie que les signaux doivent | tre mis en attente d’une certaine mani`ere si l’on veut ´eviter qu’ils soient perdus. Lorsqu’un signal parvient au bloc de destination, il est plac´e dans la file d’attente d’entr´ee du processus de r´eception. La s´emantique du LDS d´efinit pour chaque processus un m´ecanisme th´eorique de mise en file d’attente selon lequel le mode de s´election des signaux pour l’absorption par le processus est fond´e sur l’ordre d’arriv´ee des signaux dans ce processus. Lorsque le processus parvient à un ´etat, il re,coit un seul signal en provenance de la file d’attente. Ceci signifie que si la file d’attente n’est pas vide, le processus absorbe le premier signal qui vient de la file d’attente en question. Si cette derni`ere est vide, le processus demeure en attente dans l’´etat jusqu’`a l’arriv´ee à la file d’attente d’un signal qui est ensuite absorb´e par le processus.

La figure D-3.8.23 utilise le concept de file d’attente pour expliquer le fonctionnement d’un processus LDS dans lequel les temps de transition sont diff´erents de z´ero. Il convient de noter les ´el´ements suivants:

— le concept de mise en r´eserve n’est pas appliqu´e et les signaux sont absorb´es dans l’ordre de leur arriv´ee;

— l’ordre de succession de l’arrivée des signaux est important. Si <<C>> était arrivé avant <<B>> dans la transition entre l’état 1 et l’état 2, la séquence des états aurait été 1, 2, 3 au lieu de 1, 2, 4;

Figure D-3.8.22, p. 12

— la file d’attente n’´etant pas vide lorsque le processus arrive aux ´etats 2 et 4, ce processus ne doit attendre ni dans l’un ni dans l’autre de ces ´etats;

— il n’est pas possible d’attribuer la priorit´e à un signal. Un m´ecanisme sp´ecial est pr´evu pour les communications entre services, afin que les signaux ´echang´es entre service soient trait´es avant les autres signaux (§ D.5.3).

Si les temps de transition sont nuls, chaque signal sera absorb´e au moment de son arriv´ee dans un processus, sauf si l’on a recours à la mise en r´eserve (voir le § D.3.8.4).

D.3.8.3.3 *R´eception des signaux qui ne se pr´esentent pas normalement*

Dans chacun des ´etats, tous les signaux possibles doivent | tre indiqu´es implicitement ou explicitement. Des exceptions (signaux inattendus mais th´eoriquement possibles, signaux non d´efinis ou logiquement faux à un endroit donn´e, etc.) peuvent se pr´esenter dans presque tous les ´etats. Normalement, l’auteur n’indique pas ces possibilit´es; il s’ensuit qu’un tel signal sera ´elimin´e s’il se pr´esente. Si toutefois l’auteur tient à faire figurer les exceptions dans son diagramme, il doit pr´evoir pour tous les ´etats une entr´ee suppl´ementaire.

Une autre possibilit´e consiste à profiter des apparitions multiples d’un ´etat portant le symbole <<tous>> (*). Par exemple, si le signal A _RACCROCHE peut | tre re,cu dans tous les ´etats et si les actions post´erieures sont identiques, on peut recourir à la m´ethode indiqu´ee à la figure D-3.8.24.

D.3.8.3.4 *Arriv´ee simultan´ee de signaux*

La Recommandation Z.100 (§ 2) pr´evoit que les signaux peuvent parvenir simultan´ement à un processus et pr´ecise qu’ils seront plac´es dans un ordre arbitraire.

Si un usager met au point un processus capable de recevoir des signaux simultan´es, il doit veiller à ce que l’ordre d’arriv´ee ne bouleverse pas le fonctionnement escompt´e du processus.

Le LDS ne préconise pas de priorité parmi les signaux; ainsi, en cas d'arrivée simultanée de signaux, l'un d'eux est choisi arbitrairement. A noter cependant que les signaux pour communications entre services sont toujours traités les premiers (§ D.5.3).

Si plusieurs signaux sont disponibles au moment de l'entrée du processus dans un état, un seul signal est présenté au processus puis reconnu comme une entrée. Selon la sémantique du LDS, les autres signaux sont en fait retenus.

Figure D-3.8.23, p. 13

Figure D-3.8.24, p. 14

D.3.8.3.5 *Identification de l'émetteur*

Chaque signal est porteur de la valeur d'instance du processus (PId) du processus émetteur. Lorsqu'un signal est absorbé, la valeur PId du processus émetteur peut être obtenue au moyen de l'expression SENDER. On trouvera dans la figure D-3.8.25 un exemple d'emploi de cette variable.

Figure D-3.8.25, p. 15

D.3.8.4 *Mises en réserve*

Le concept de mise en réserve permet de différer l'absorption d'un signal jusqu'à ce qu'un ou plusieurs signaux ultérieurs aient été absorbés. Comme l'indique le § D.3.8.3, les signaux sont absorbés dans l'ordre dans lequel ils se présentent, sauf en cas d'emploi du concept de mise en réserve.

L'on peut faire appel au concept de mise en réserve afin de simplifier les processus lorsque l'ordre relatif d'arrivée de certains signaux n'a pas d'importance et que leur ordre effectif d'arrivée est indéterminé.

Dans chaque état, chaque signal est traité comme suit:

— il est représenté comme un symbole d'entrée ou,

— il est représenté comme un symbole de mise en réserve ou,

— il est, par convention, couvert par une entrée implicite aboutissant à une transition nulle implicite.

Le fonctionnement du mécanisme implicite de mise en file d'attente décrit dans le § D.3.8.3 s'applique également au concept de mise en réserve. A leur arrivée, les signaux sont placés dans la file d'attente et lorsque le processus atteint un état donné, les signaux qui se trouvent dans la file d'attente sont examinés un à un dans l'ordre de leur arrivée. Un signal couvert par un symbole d'entrée explicite ou implicite est absorbé et la transition qui s'y rapporte est exécutée. Un signal représenté dans un symbole de mise en réserve n'est pas absorbé et reste dans la file d'attente dans la même position séquentielle; le signal suivant de la file d'attente est considéré. Aucune transition ne suit un symbole de mise en réserve.

En LDS/PR, la construction de mise en réserve est exprimée à l'aide du mot-clé SAVE suivi d'une liste d'identificateurs de signaux. On trouvera un exemple simple d'énoncés de MISE EN RÉSERVE dans la figure D-3.8.26.

Figure D-3.8.26 [T15.100]   (a traiter comme tableau MEP), p. 16

En LDS/GR, le concept de mise en réserve est représenté à l'aide du symbole de mise en réserve contenant les identificateurs de signaux.

Comme dans le cas des entrées, une <<notation avec astérisque>> peut servir à représenter la mise en réserve de tous les signaux (valides pour ce processus) qui ne sont pas explicitement mentionnés dans cet état.

La figure D-3.8.27 représente un exemple d'un processus LDS qui comporte un symbole de mise en réserve. Il convient de noter que les signaux S et R sont consommés dans l'ordre R, S, c'est-à-dire dans l'ordre inverse de leur réception. Un symbole de mise en réserve unique peut servir à mettre un signal en réserve tant que le processus se trouve dans l'état dans lequel le symbole appara | t; ce signal est mis en réserve pour la durée de la transition vers le prochain état. Dans le prochain état, le signal sera absorbé par l'intermédiaire d'une entréeexplicite ou implicite (voir la figure D-3.8.27) sauf dans les cas suivants: lorsque le symbole de mise en réserve comportant le nom du signal est répété ou lorsque dans la file d'attente implicite, il existe avant lui, un autre signal de sauvegarde disponible pour absorption (voir la figure D-3.8.28).

Figure D-3.8.27, p. 17

Figure D-3.8.28, p. 18

Un signal mis en réserve n'est mis à disposition d'un processus que par l'intermédiaire d'un symbole d'entrée correspondant (explicite ou implicite). Aucune question relative à un signal mis en réserve ne peut être posée dans une décision avant la reconnaissance de ce signal comme une entrée; de même les données qui lui sont associées ne sont pas disponibles.

Dans un état où plusieurs signaux doivent être mis en réserve, on peut attribuer un symbole de mise en réserve à chaque signal ou on peut les représenter tous à l'intérieur du même symbole de mise en réserve. Si plusieurs signaux doivent être mis en réserve, la sémantique du symbole de mise en réserve exige que l'ordre de leur arrivée soit préservé.

Un troisième exemple de l'utilisation de la notion de mise en réserve est donné dans la figure D-3.8.29 et la figure D-3.8.30 décrit le fonctionnement du mécanisme de formation de la file d'attente.

L'utilisation du symbole de mise en réserve peut simplifier les diagrammes. Ainsi, en mettant un signal en réserve, l'on peut éviter de le recevoir et de devoir mettre en mémoire ses données associées jusqu'à l'état suivant.

Bien que le symbole de mise en réserve puisse être utilisé à chaque niveau de description, il y aurait peut-être lieu, au niveau inférieur, de décrire le mécanisme effectif qui permet cette mise en réserve.

Sans un minimum de précautions dans l'emploi de la mise en réserve, la file d'attente des signaux mis en réserve peut augmenter considérablement ou garder des signaux en mémoire trop longtemps, de sorte qu'un signal ancien peut être absorbé lorsqu'un signal récent est demandé.

Le LDS ne prévoit pas de limite à la longueur de la file d'attente, ce qui peut poser des problèmes de mise en oeuvre.

Figure D-3.8.29, p. 19

Figure D-3.8.30, p. 20

D.3.8.5 *Condition de validation et signaux continus*

Les conditions de validation permettent une réception conditionnelle de signaux fondée sur la condition de validation spécifiée. Le signal est reçu et la transition interprétée si la condition est vraie. En cas de condition fausse, le signal est mis en réserve et le processus ne change pas d'état jusqu'à ce qu'un autre signal arrive ou que la condition devienne vraie. L'exemple de la figure D-3.8.31 illustre ceci. Lorsque P1 passe à l'état S1, la condition (c'est-à-dire de savoir si IMPORT (X,P2) est égal à 10) est évaluée. Si elle est vraie, le signal B peut être reçu. Sinon, le signal B est mis en réserve. Dans cet exemple, A arrive et provoque une transition vers S2. Pendant la transition, X passe à la valeur 11 et P2 exporte sa nouvelle valeur; alors, la condition liée au signal B dans l'état S2 est vraie. Etant donné que B est le premier signal de la file d'attente, la transition qui suit est exécutée et le processus prend fin à l'état S3.

Certaines caractéristiques des conditions de validation sont importantes:

1) la condition de validation est testée lorsque le processus arrive à un état; il est alors continuellement contrôlé tandis que le processus reste dans cet état. Ainsi, si la valeur exportée de X avait passé de 9 à 11 puis à 12 pendant la transition qui faisait suite à la réception de A, le processus serait resté à S2;

2) les conditions de validation peuvent être fondées sur des variables locales et/ou toute construction de langage qui peut être comprise dans une expression (par exemple, IMPORT (IMPORTATION), VIEW (VUE), NOW (MAINTENANT));

3) alors qu'il est possible d'employer plus d'une condition par état, l'emploi de plus d'une condition de validation pour le même signal n'est pas autorisé. Ainsi, la condition indiquée dans la figure D-3.8.32 n'est pas autorisée. Si un signal donné exige des conditions multiples, il est possible de les combiner en une expression booléenne ainsi que le montre la figure D-3.8.33.

On peut évaluer les conditions de validation plusieurs fois et dans un ordre quelconque, de sorte que l'usager doit être vigilant si les expressions ont des effets secondaires réciproques (par exemple IMPORT et SENDER combinés).

Il faut noter en outre que le signal spécifié dans la condition de validation ne peut influencer la valeur booléenne de la condition car ses valeurs transportées ne sont pas affectées avant l'absorption du signal. A titre d'exemple, les énoncés:

INPUT x(A) PROVIDED (A=5);... INPUT y PROVIDED(SENDER)=pid1);

prêtent à confusion car les valeurs de A et de SENDER dans les conditions correspondent à la situation telle qu'elle était avant l'absorption du signal.

Les signaux continus ont les mêmes propriétés fondamentales que la condition de validation, excepté le fait qu'ils ne sont accompagnés d'aucun signal. Ainsi, en l'absence de signaux dans la file d'attente susceptibles de provoquer une transition à leur entrée dans l'état, les signaux continus sont contrôlés; si l'un d'entre eux est vrai, la transition qui le suit est exécutée. L'exemple de la figure D-3.8.34 en donne l'illustration. A l'origine, le processus se trouve à l'état S1 et la valeur exportée de X est 9. En arrivant, le signal A provoque la transition vers l'état S2. Pendant la transition, X prend la valeur 11. Vu qu'aucun autre signal ne se trouve dans la file d'attente, la transition vers l'état S3 s'accomplit.

L'on trouvera ci-dessous certaines caractéristiques importantes des signaux continus:

1) de même que pour les conditions de validation, la valeur de la condition n'est contrôlée qu'à l'arrivée à un état;

2) les signaux continus multiples sont autorisés pour chaque état. Lorsque plusieurs signaux continus sont liés à un état, le signal continu ayant le rang de priorité le plus élevé (le numéro le plus bas) sera traité le premier. Deux signaux continus ne peuvent avoir le même rang de priorité. Le signal continu est toujours moins prioritaire que tout autre signal. Ceci est de nouveau dû au système sous-jacent du LDS: toutefois, la représentation à l'aide de modèles des signaux continus en LDS (emploi des signaux émis pendant l'exportation de variables), permet le recours à des priorités pour les signaux continus: ceci est d'ailleurs nécessaire afin d'éviter toute ambiguïté en cas de présence de plusieurs signaux continus. La figure D-3.8.35 en donne une illustration. Le processus commence à l'état S1 et ses variables locales X et Y ont respectivement les valeurs 10 et 11. Etant donné que les deux signaux continus sont vrais, celui qui a la plus haute priorité (rang de priorité le plus bas) est choisi et la transition vers l'état S2 s'accomplit. En S2, la condition de Y n'est plus vraie, et bien que

la priorit´e du signal continu de X soit inf´erieure à celle de Y, la transition qui le suit est ex´ecut´ee et le processus parvient à l’´etat S3;

3) lorsque la transition d’un signal continu a une suite, l’expression SENDER (ÉMETTEUR) retourne la m |me valeur de SELF.

Figure D-3.8.31, p. 21

Figure D-3.8.32, p. 22

Figure D-3.8.33, p. 23

Figure D-3.8.34, p. 24

Figure D-3.8.35, p. 25

D.3.8.6 *Sorties*

Une sortie est l’´emission d’un signal d’un processus vers un autre ou vers lui-m | me. Etant donn´e que le contr | le de la r´eception et de l’absorption du signal est associ´e au processus de r´eception (voir le § D.3.8.3), les r`egles s´emantiques se rapportant directement aux symboles de sorties sont relativement simples. Du point de vue du processus d’´emission, une sortie peut souvent | tre consid´er´ee comme une action instantan´ee qui, une fois achev´ee, n’a aucun autre effet direct sur le processus d’´emission, lequel ne sera pas directement conscient du sort du signal.

Une action de sortie repr´esente l’´emission d’un signal et l’association de valeurs s’il en existe. Il est possible d’associer des valeurs à un signal de sortie en les pla,cant entre parenth`eses ou en mettant des expressions ayant des valeurs entre parenth`eses (voir la figure D-3.8.37).

En LDS/PR, une action de sortie est repr´esent´ee à l’aide du mot-cl´e OUTPUT (sortie) suivi d’une liste d’identificateurs de signaux. Une liste de param`etres r´eels mis entre parenth`eses peut | tre associ´ee à chaque identificateur de signal. S’il n’existe pas en fait de param`etre r´eel dans la sortie correspondant à une sorte dans la d´efinition du signal, la variable correspondante dans l’entr´ee de r´eception aura la valeur <<ind´efinie>>.

L’instance de processus de destination doit | tre exprim´ee dans l’instruction de sortie par le mot-cl´e TO (vers) suivi d’une expression d’instance de processus. Si l’instance de processus de destination peut | tre d´etermin´ee uniquement par le contexte, la clause TO peut | tre omise. Une condition d’adressage suppl´ementaire peut | tre fournie dans l’´enonc´e de sortie à l’aide du mot-cl´e VIA suivi d’une liste d’acheminement de signaux et d’identificateurs de canaux.

La figure D-3.8.36 donne quelques exemples valables d’´enonc´es de sortie.

Figure D-3.8.36 [T16.100] (a traiter comme tableau MEP), p. 26

En LDS/GR, une sortie est repr´esent´ee à l'aide d'un symbole de sortie contenant la sp´ecification de signaux, de param`etres r´eels et, en option, de destination et/ou de construction VIA.

Chaque sortie doit | tre dirig´ee vers une instance de processus donn´ee. Etant donn´e qu'il est g´en´eralement impossible de conna | tre l'instance de processus de tout processus au moment de l'´elaboration d'une sp´ecification, la m´ethode normale pour diriger les signaux consiste à employer une variable ou une expression dans le mot-cl´e TO (VERS). On trouvera dans les figures D-3.8.38, D-3.8.39 et D-3.8.40 des exemples. Dans la figure D-3.8.38, le param`etre de processus <<out _to>> prend la valeur d'une instance de processus lors de la cr´eation du processus. Le r | le de <<out _to>> au sein du processus est d'assurer la liaison entre le processus en question et le processus auquel il est connect´e. Il convient de veiller lors de la conception du syst`eme, à ce que le type de processus indiqu´e par <<out _to>> puisse recevoir les signaux qui sont ´emis. Dans la figure D-3.8.39, l'expression pr´ed´efinie SENDER sert à renvoyer un signal au processus qui a ´emis le signal re,cu peu avant. Dans la figure D-3.8.40, le signal est envoy´e vers le descendant le plus r´ecent du processus.

Figure D-3.8.37, p. 27

Figure D-3.8.38 [T17.100] (a traiter comme tableau MEP), p. 28

Figure D-3.8.39, p. 29

Figure D-3.8.40, p. 30

D.3.8.7 *T | che*

Dans une transition, une t | che sert à repr´esenter à l'aide d'un texte informel des op´erations sur des variables ou une op´eration sp´eciale.

En LDS/PR, une t | che est repr´esent´ee par le mot-cl´e TASK (T | CHE) suivi d'une liste d'instructions ou de textes informels s´epar´es par des virgules et se terminant par un point-virgule. Une instruction d'une t | che peut consister simplement en une affectation. Un texte informel consiste en une phrase d´elimit´ee par des apostrophes. On trouvera dans la figure D-3.8.41 des exemples de t | ches valables en LDS/PR.

Figure D-3.8.41 [T18.100] (a traiter comme tableau MEP), p. 31

En LDS/GR, une tâche se compose d'un symbole de tâche contenant la liste des instructions ou des textes informels.

Les usagers du LDS peuvent parfois éprouver de la difficulté à décider si un aspect du système défini doit être représenté par une tâche ou un processus distinct. Considérons l'exemple du processus représenté dans la figure D-3.8.42: l'action <<connecter-trajet-de-commutation>> doit-elle être représentée par une tâche ou par un processus distinct? Si l'on n'a pas identifié un processus distinct de commande de trajet de commutation, il serait indiqué d'utiliser le symbole tâche [voir la figure D-3.8.42 a)]. Si on a identifié un processus distinct de commande de trajet de commutation, on doit utiliser les signaux qui communiquent avec le processus de commande [voir la figure D-3.8.42 b)].

Figure D-3.8.42, p. 32

D.3.8.8 *Décisions*

Une décision est une action qui se produit au cours d'une transition et qui correspond à une question concernant la valeur d'une expression au moment de l'exécution de l'action; le processus a le choix entre deux ou plusieurs trajets, ce choix étant déterminé par la réponse consécutive à la décision. Les auteurs des diagrammes LDS doivent veiller à ce que les processus soient définis de manière qu'ils ne puissent tenter d'exécuter des décisions pour lesquelles des réponses (ou les données) ne sont pas disponibles; de telles décisions rendraient le diagramme tout à fait incorrect et entraîneraient une confusion considérable.

La question à laquelle correspond une décision peut être une expression ou un texte informel. Les réponses apportées par une décision sont représentées par une ou plusieurs valeurs possibles obtenues par l'évaluation de l'expression de la question ou par un ou plusieurs textes informels. Si la question ou l'une des réponses est informelle, toute la décision est informelle. Des réponses différentes sont séparées par des virgules. Les valeurs sont représentées par des expressions constantes, par des expressions constantes ayant un opérateur comme préfixe ou par des gammes dont les limites supérieures et inférieures sont des expressions constantes. Les valeurs de réponse doivent être du même type que l'expression contenue dans la question.

Il est possible d'indiquer explicitement certaines réponses et de grouper toutes les autres réponses possibles en employant le mot-clé ELSE (AUTRE).

En LDS/PR, la décision est représentée par le mot-clé DÉCISION suivi par la spécification de la question et la liste des réponses possibles, chacune étant associée à la transition correspondante. Les réponses sont indiquées entre parenthèses. L'ensemble des transitions sortantes est délimité par le mot-clé ENDDECISION (FIN DE DÉCISION) placé à la fin (voir la figure D-3.8.43).

Figure D-3.8.43 [T19.100] (à traiter comme tableau MEP), p. 33

On trouvera certains exemples de décisions dans la figure D-3.8.44.

Figure D-3.8.44 [T20.100] (à traiter comme tableau MEP), p. 34

Toutes les transitions se terminent par le mot-clé ENDECISION (FIN DE DÉCISION). Les décisions qui ne se terminent pas par un énoncé terminal (c'est-à-dire jonction, état suivant, arrêt) continuent dans l'énoncé qui suit le mot ENDECISION, comme indiqué dans les deux branches équivalentes de la figure D-3.8.45.

Figure D-3.8.45, p. 35

L'instruction de décision peut en outre servir à former la structure IF-THEN (SI-ALORS), la structure DO-WHILE (FAIRE-PENDANT) et la structure LOOP-UNTIL (BOUCLE-JUSQU'A) comme en programmation structurée.

En LDS/GR, une décision est représentée à l'aide d'un symbole de décision contenant le texte de la question. Le symbole doit avoir deux branches ou plus associées aux réponses correspondantes. Chaque réponse doit être placée à la droite ou au sommet de la branche correspondante ou au-dessus de la branche qui interrompt la ligne de liaison. En LDS/GR, les parenthèses servant à délimiter les réponses sont facultatives mais il est suggéré de les utiliser pour éviter tout malentendu.

On trouvera certains exemples de décisions en LDS/GR dans la figure D-3.8.46.

Figure D-3.8.46, p. 36

Si une r´eponse ram`ene à la d´ecision de la m | me transition, il convient d'ex´ecuter certaines actions qui influencent la question ayant trait à la d´ecision. Toutefois, m | me si cette r`egle est appliqu´ee, des boucles infinies peuvent appara | tre, comme indiqu´e dans la figure D-3.8.47. Il faut donc toujours faire attention lorsque des r´eponses se r´ef`erent à une d´ecision de la m | me transition.

Figure D-3.8.47, p. 37

Des d´ecisions peuvent | tre prises à l'aide de toute valeur existant dans le processus et notamment:

— des valeurs re,cues par une entr´ee;

— des valeurs transmises en tant que param`etres effectifs au moment de la cr´eation du processus;

— des valeurs partag´ees.

L'expression qui figure dans la question peut comprendre des constantes de l'un quelconque des types de valeurs susmentionn´es.

D.3.8.9 *Branchements et connecteurs*

Les branchements permettent de transf´erer la commande d'un point à un autre d'un corps de processus (ainsi qu'`a l'int´erieur d'un corps de proc´edure ou d'un corps de service).

En LDS/PR, les branchements sont ´equivalentsà des ´enonc´es <<GO TO>>. Des ´etiquettes sont utilis´ees comme points d'introduction associ´es aux instructions, comme indiqu´e dans la figure D-3.8.48. A l'int´erieur d'un corps de processus (ou d'un corps de proc´edure), il est impossible de transf´erer la commande (et par cons´equent les ´etiquettes associ´ees) au type instructions indiqu´e dans la figure D-3.8.49. Les ´etiquettes demeurent toujours localis´ees dans un processus; il est donc impossible de transf´erer la commande d'un processus à un autre à l'aide d'un branchement.

Figure D-3.8.48, p. 38

Figure D-3.8.49 [T21.100] (à traiter comme tableau MEP), p. 39

En LDS/GR, les branchements correspondent aux connecteurs (de sortie et d'entrée). Elles peuvent être utilisées pour subdiviser les programmes, en cas de manque de place, ou pour éviter le croisement de lignes de liaison qui enlèverait de leur clarté aux diagrammes. En outre, il est généralement préférable, lorsque l'on trace un diagramme en LDS, que la liaison se dirige du haut au bas de la page.

En GR, toute ligne de liaison peut être interrompue par une paire de connecteurs associés; on admet alors que la liaison se dirige du connecteur de sortie au connecteur d'entrée. Chaque symbole de connecteur contient un nom, associé aux connecteurs portant le même nom. Il existe un seul connecteur d'entrée pour chaque nom mais il peut y avoir un ou plusieurs connecteurs de sortie.

En GR, il est souhaitable que la page de référence se rapportant au connecteur d'entrée approprié soit spécifiée pour le connecteur de sortie et que la ou les références de pages relatives aux connecteurs de sortie appropriés devraient être données pour le connecteur d'entrée (voir l'exemple de la figure D-3.8.50).

Figure D-3.8.50, p. 40

D.3.9 *Procédures*

En LDS, les procédures sont similaires aux procédures du CHILL et d'autres langages de programmation. Elles visent à:

a) permettre de structurer un processus en plusieurs niveaux de précision;

b) conserver la compacité des spécifications en permettant de représenter comme un seul élément un assemblage complexe d'éléments qui peuvent être considérés isolément;

c) permettre de définir et d'utiliser à plusieurs reprises les assemblages d'éléments utilisés.

Une définition de procédure ne peut exister que dans une définition de processus, dans une définition de service ou une définition de procédure et, par conséquent, une procédure n'est visible que pour le processus ou la procédure dans lesquels elle est définie.

Une définition de procédure se compose des éléments suivants (dont certains sont facultatifs):

— nom de la procédure;

— paramètres formels de procédure: liste de noms de variables associées à leur sorte. Ces paramètres servent à transférer l'information de la procédure ou à partir de celle-ci au moment de l'appel. Les paramètres de procédure peuvent être passés par valeur (paramètre IN) ou par référence (paramètre IN/OUT). Si un paramètre est passé par valeur, la spécification du paramètre formel définit une variable locale à la procédure; s'il est passé par référence, la spécification définit un synonyme pour la variable;

— définitions de procédure: procédures qui peuvent être appelées tout comme la procédure proprement dite;

— définitions de données: spécification de types de données locales à la procédure;

— définitions de variables: variables locales à la procédure;

— corps de procédure: spécification du comportement effectif de la procédure pour ce qui est des états et des actions (comme pour le corps de processus).

On trouvera dans la figure D-3.9.1 un exemple partiel de définition de procédure en LDS/PR (les mots-clés du langage sont écrits en majuscules). A noter que les paramètres formels sans attribut explicite ont un attribut implicite IN (var5 dans la figure).

Figure D-3.9.1 [T22.100] (à traiter comme tableau MEP), p. 41

D.3.9.1 *Corps de procédure*

Le corps de procédure présente de grandes similitudes avec le corps de processus; toutefois les différences sont les suivantes:

— la procédure termine son interprétation avec une indication <<retour>> au lieu d'une indication <<arrêt>>. En LDS/PR, l'énoncé de retour est représenté par le mot-clé RETURN;

— en LDS/GR, le symbole de début de procédure est légèrement différent du symbole de début de processus.

Les symboles de début et de retour de procédure sont indiqués dans le résumé du LDS/GR.

Une procédure peut utiliser la construction avec branchements mais seulement pour se référer à une étiquette interne. Un branchement peut ne pas être utilisé ou être utilisé pour accéder à une procédure de l'extérieur ou quitter celle-ci.

En LDS/GR, une définition de procédure est représentée par un diagramme de procédure très similaire au diagramme de processus. Le diagramme de procédure comprend les éléments suivants:

— un symbole de cadre facultatif: symbole de forme rectangulaire contenant tous les autres symboles;

— l'en-tête de procédure: le mot-clé PROCEDURE suivi du nom de la procédure et de la spécification des paramètres formels de procédure. Généralement, l'en-tête de procédure est placée dans l'angle supérieur gauche du cadre ou, s'il n'y a pas de cadre, dans l'angle supérieur gauche du support sur lequel le diagramme est tracé;

— une numérotation de pages facultatives (à placer dans l'angle supérieur droit);

— des symboles de texte: dans le cas d'un diagramme de procédure, un symbole de texte peut être utilisé pour contenir la spécification de définition de paramètres formels, de données et de variables;

— références de procédure: symboles de procédure, contenant chacun un nom de procédure représentant une procédure locale définie séparément;

— des diagrammes de proc´edure: permettant de d´efinir directement les proc´edures locales;

— la zone graphique de proc´edure: sp´ecification du comportement de la proc´edure pour ce qui est du d´ebut, des ´etats, des entr´ees, des sorties, des t | ches... et des arcs orient´es.

Dans la figure D-3.9.2, on trouvera un exemple de d´efinition de proc´edure en LDS/GR. La proc´edure portant la r´ef´erence <<TERM\u(emP>> dans l'exemple est locale à la proc´edure d'appel.

Comme indiqu´e dans le cas des diagrammes de proc´edure (§ D.3.8), si une seule page ne suffit pas à contenir un diagramme de proc´edure, celui-ci peut | tre repr´esent´e sur plusieurs pages, avec r´ep´etition du symbole de cadre à la suite de l'en-t | te et du num´ero de page.

Figure D-3.9.2, p. 42

D.3.9.2 *Appel de procédure*

Les appels de procédure peuvent se produire chaque fois qu'une tâche et autorisée dans un graphe de processus ou de procédure. En un sens, une procédure peut être interprétée comme une tâche, avec les exceptions suivantes:

1) une procédure peut contenir des états et, si tel est le cas, recevoir des signaux;

2) une procédure peut émettre des signaux. L'instance de processus d'origine est celle qui a appelé la procédure.

Lorsqu'une procédure est appelée, son environnement est créé et l'interprétation de la procédure commence. Elle se poursuit jusqu'à ce que l'on ait atteint l'indication RETURN (retour). Pendant l'interprétation de la procédure, tous les signaux adressés au processus sont soit mis en réserve implicitement soit traités explicitement par la procédure. La procédure n'a pas sa propre file d'attente mais utilise celle du processus qui l'a appelée.

En LDS/PR, un appel de procédure est représenté par le mot-clé CALL suivi de l'identificateur de procédure et de la liste de paramètres réels mise entre parenthèses. Si un paramètre n'est pas donné, il convient de l'indiquer par deux virgules consécutives. Dans ce cas, le paramètre formel correspondant a la valeur <<indéfinie>>. A noter aussi que la déclaration de IN ou IN/OUT est faite dans la définition de procédure, de sorte qu'elle ne doit pas être répétée par l'énoncé d'appel. On trouvera certains exemples d'appel en LDS/PR dans la figure D-3.9.3.

Figure D-3.9.3 [T23.100] (à traiter comme tableau MEP), p. 43

En LDS/GR, un appel de procédure est représenté à l'aide d'un symbole d'appel de procédure contenant le nom de procédure et la liste des paramètres réels mise entre paranthèses. On trouvera un exemple d'appel en LDS/GR dans la figure D-3.9.2.

Blanc