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.

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 ´t´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 ´t´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 ´t´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 ´etats 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 ´t´e ´emis), à moins que cette information n'ait ´t´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.

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 plac´es entre parenth`eses et s´epar´es par des virgules (voir la figure D-3.8.16). Si une ou plusieurs valeurs de signal sont rejet´ees, les variables correspondantes font d´efaut, ce qui est indiqu´e par deux ou plusieurs virgules cons´ecutives (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'arriv´ee des signaux est important. Si <<C>> ´etait arriv´e avant <<B>> dans la transition entre l'´etat 1 et l'´etat 2, la s´equence des ´etats aurait ´et´e 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_RACCROCHÉ 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 pr´econise pas de priorit´e parmi les signaux; ainsi, en cas d'arriv´ee simultan´ee de signaux, l'un d'eux est choisi arbitrairement. A noter cependant que les signaux pour communications entre services sont toujours trait´es les premiers (§ D.5.3).

Si plusieurs signaux sont disponibles au moment de l'entr´ee du processus dans un ´etat, un seul signal est pr´esent´e au processus puis reconnu comme une entr´ee. Selon la s´emantique 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´´emetteur*

Chaque signal est porteur de la valeur d'instance du processus (PId) du processus ´emetteur. Lorsqu'un signal est absorb´e, la valeur PId du processus ´emetteur peut |tre 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 r´eserve*

Le concept de mise en r´eserve permet de diff´erer l'absorption d'un signal jusqu'`a ce qu'un ou plusieurs signaux ult´erieurs aient ´et´e absorb´es. Comme l'indique le § D.3.8.3, les signaux sont absorb´es dans l'ordre dans lequel ils se pr´esentent, sauf en cas d'emploi du concept de mise en r´eserve.

L'on peut faire appel au concept de mise en r´eserve afin de simplifier les processus lorsque l'ordre relatif d'arriv´ee de certains signaux n'a pas d'importance et que leur ordre effectif d'arriv´ee est ind´etermin´e.

Dans chaque ´etat, chaque signal est trait´e comme suit:

— il est repr´esent´e comme un symbole d'entr´ee ou,

— il est repr´esent´e comme un symbole de mise en r´eserve ou,

— il est, par convention, couvert par une entr´ee implicite aboutissant à une transition nulle implicite.

Le fonctionnement du m´ecanisme implicite de mise en file d'attente d´ecrit dans le § D.3.8.3 s'applique ´egalement au concept de mise en r´eserve. A leur arriv´ee, les signaux sont plac´es dans la file d'attente et lorsque le processus atteint un ´etat donn´e, les signaux qui se trouvent dans la file d'attente sont examin´es un à un dans l'ordre de leur arriv´ee. Un signal couvert par un symbole d'entr´ee explicite ou implicite est absorb´e et la transition qui s'y rapporte est ex´ecut´ee. Un signal repr´esent´e dans un symbole de mise en r´eserve n'est pas absorb´e et reste dans la file d'attente dans la m|me position s´equentielle; le signal suivant de la file d'attente est consid´er´e. Aucune transition ne suit un symbole de mise en r´eserve.

En LDS/PR, la construction de mise en r´eserve est exprim´ee à l'aide du mot-cl´e SAVE suivi d'une liste d'identificateurs de signaux. On trouvera un exemple simple d'´enonc´es de MISE EN RÉSERVE dans la figure D-3.8.26.

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

En LDS/GR, le concept de mise en r´eserve est repr´esent´e à l'aide du symbole de mise en r´eserve contenant les identificateurs de signaux.

Comme dans le cas des entr´ees, une <<notation avec ast´erisque>> peut servir à repr´esenter la mise en r´eserve de tous les signaux (valides pour ce processus) qui ne sont pas explicitement mentionn´es dans cet ´etat.

La figure D-3.8.27 repr´esente un exemple d'un processus LDS qui comporte un symbole de mise en r´eserve. Il convient de noter que les signaux S et R sont consomm´es dans l'ordre R, S, c'est-`a-dire dans l'ordre inverse de leur r´eception. Un symbole de mise en r´eserve unique peut servir à mettre un signal en r´eserve tant que le processus se trouve dans l'´etat dans lequel le symbole appara | t; ce signal est mis en r´eserve pour la dur´ee de la transition vers le prochain ´etat. Dans le prochain ´etat, le signal sera absorb´e par l'interm´ediaire d'une entr´eeexplicite ou implicite (voir la figure D-3.8.27) sauf dans les cas suivants: lorsque le symbole de mise en r´eserve comportant le nom du signal est r´ep´et´e 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 r´eserve n'est mis à disposition d'un processus que par l'interm´ediaire d'un symbole d'entr´ee correspondant (explicite ou implicite). Aucune question relative à un signal mis en r´eserve ne peut |tre pos´ee dans une d´ecision avant la reconnaissance de ce signal comme une entr´ee; de m|me les donn´ees qui lui sont associ´ees ne sont pas disponibles.

Dans un ´etat où plusieurs signaux doivent |tre mis en r´eserve, on peut attribuer un symbole de mise en r´eserve à chaque signal ou on peut les repr´esenter tous à l'int´erieur du m|me symbole de mise en r´eserve. Si plusieurs signaux doivent |tre mis en r´eserve, la s´emantique du symbole de mise en r´eserve exige que l'ordre de leur arriv´ee soit pr´eserv´e.

Un troisi`eme exemple de l'utilisation de la notion de mise en r´eserve est donn´e dans la figure D-3.8.29 et la figure D-3.8.30 d´ecrit le fonctionnement du m´ecanisme de formation de la file d'attente.

L'utilisation du symbole de mise en r´eserve peut simplifier les diagrammes. Ainsi, en mettant un signal en r´eserve, l'on peut ´eviter de le recevoir et de devoir mettre en m´emoire ses donn´ees associ´ees jusqu'`a l'´etat suivant.

Bien que le symbole de mise en r´eserve puisse |tre utilis´e à chaque niveau de description, il y aurait peut-|tre lieu, au niveau inf´erieur, de d´ecrire le m´ecanisme effectif qui permet cette mise en r´eserve.

Sans un minimum de pr´ecautions dans l'emploi de la mise en r´eserve, la file d'attente des signaux mis en r´eserve peut augmenter consid´erablement ou garder des signaux en m´emoire trop longtemps, de sorte qu'un signal ancien peut |tre absorb´e lorsqu'un signal r´ecent est demand´e.

Le LDS ne pr´evoit pas de limite à la longueur de la file d'attente, ce qui peut poser des probl`emes 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 r´eception conditionnelle de signaux fond´ee sur la condition de validation sp´ecifi´ee. Le signal est re,cu et la transition interpr´et´ee si la condition est vraie. En cas de condition fausse, le signal est mis en r´eserve et le processus ne change pas d'´etat jusqu'`a 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'´etat S1, la condition (c'est-`a-dire de savoir si IMPORT (X,P2) est ´egal à 10) est ´evalu´ee. Si elle est vraie, le signal B peut | tre re,cu. Sinon, le signal B est mis en r´eserve. 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 li´ee au signal B dans l'´etat S2 est vraie. Etant donn´e que B est le premier signal de la file d'attente, la transition qui suit est ex´ecut´ee et le processus prend fin à l'´etat S3.

Certaines caract´eristiques des conditions de validation sont importantes:

1) la condition de validation est test´ee lorsque le processus arrive à un ´etat; il est alors continuellement contr | l´e tandis que le processus reste dans cet ´etat. Ainsi, si la valeur export´ee de X avait pass´e de 9 à 11 puis à 12 pendant la transition qui faisait suite à la r´eception de A, le processus serait rest´e à S2;

2) les conditions de validation peuvent | tre fond´ees sur des variables locales et/ou toute construction de langage qui peut | tre 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 ´etat, l'emploi de plus d'une condition de validation pour le m | me signal n'est pas autoris´e. Ainsi, la condition indiqu´ee dans la figure D-3.8.32 n'est pas autoris´ee. Si un signal donn´e exige des conditions multiples, il est possible de les combiner en une expression bool´eenne ainsi que le montre la figure D-3.8.33.

On peut ´evaluer les conditions de validation plusieurs fois et dans un ordre quelconque, de sorte que l'usager doit | tre vigilant si les expressions ont des effets secondaires r´eciproques (par exemple IMPORT et SENDER combin´es).

Il faut noter en outre que le signal sp´ecifi´e dans la condition de validation ne peut influencer la valeur bool´eenne de la condition car ses valeurs transport´ees ne sont pas affect´ees avant l'absorption du signal. A titre d'exemple, les ´enonc´es:

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

pr | tent à confusion car les valeurs de A et de SENDER dans les conditions correspondent à la situation telle qu'elle ´etait avant l'absorption du signal.

Les signaux continus ont les m | mes propri´et´es fondamentales que la condition de validation, except´e le fait qu'ils ne sont accompagn´es d'aucun signal. Ainsi, en l'absence de signaux dans la file d'attente susceptibles de provoquer une transition à leur entr´ee dans l'´etat, les signaux continus sont contr | l´es; si l'un d'entre eux est vrai, la transition qui le suit est ex´ecut´ee. L'exemple de la figure D-3.8.34 en donne l'illustration. A l'origine, le processus se trouve à l'´etat S1 et la valeur export´eede X est 9. En arrivant, le signal A provoque la transition vers l'´etat 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'´etat S3 s'accomplit.

L'on trouvera ci-dessous certaines caract´eristiques importantes des signaux continus:

1) de m | me que pour les conditions de validation, la valeur de la condition n'est contr | l´ee qu'`a l'arriv´ee à un ´etat;

2) les signaux continus multiples sont autoris´es pour chaque ´etat. Lorsque plusieurs signaux continus sont li´es à un ´etat, le signal continu ayant le rang de priorit´e le plus ´elev´e (le num´ero le plus bas) sera trait´e le premier. Deux signaux continus ne peuvent avoir le m | me rang de priorit´e. Le signal continu est toujours moins prioritaire que tout autre signal. Ceci est de nouveau d | au syst`eme sous-jacent du LDS: toutefois, la repr´esentation à l'aide de mod`eles des signaux continus en LDS (emploi des signaux ´emis pendant l'exportation de variables), permet le recours à des priorit´es pour les signaux continus: ceci est d'ailleurs n´ecessaire afin d'´eviter toute ambigu"it´e en cas de pr´esence de plusieurs signaux continus. La figure D-3.8.35 en donne une illustration. Le processus commence à l'´etat S1 et ses variables locales X et Y ont respectivement les valeurs 10 et 11. Etant donn´e que les deux signaux continus sont vrais, celui qui a la plus haute priorit´e (rang de priorit´e le plus bas) est choisi et la transition vers l'´etat 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] (à 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-3.8.38 [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 t | che se compose d'un symbole de t | che contenant la liste des instructions ou des textes informels.

Les usagers du LDS peuvent parfois ´eprouver de la difficult´e à d´ecider si un aspect du syst`eme d´efini doit | tre repr´esent´e par une t | che ou un processus distinct. Consid´erons l'exemple du processus repr´esent´e dans la figure D-3.8.42: l'action <<connecter-trajet-de-commutation>> doit-elle | tre repr´esent´ee par une t | che ou par un processus distinct? Si l'on n'a pas identifi´e un processus distinct de commande de trajet de commutation, il serait indiqu´e d'utiliser le symbole t | che [voir la figure D-3.8.42 a)]. Si on a identifi´e 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 *D´ecisions*

Une d´ecision 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'ex´ecution de l'action; le processus a le choix entre deux ou plusieurs trajets, ce choix ´etant d´etermin´e par la r´eponse cons´ecutive à la d´ecision. Les auteurs des diagrammes LDS doivent veiller à ce que les processus soient d´efinis de mani`ere qu'ils ne puissent tenter d'ex´ecuter des d´ecisions pour lesquelles des r´eponses (ou les donn´ees) ne sont pas disponibles; de telles d´ecisions rendraient le diagramme tout à fait incorrect et entra | neraient une confusion consid´erable.

La question à laquelle correspond une d´ecision peut | tre une expression ou un texte informel. Les r´eponses apport´ees par une d´ecision sont repr´esent´ees par une ou plusieurs valeurs possibles obtenues par l'´evaluation de l'expression de la question ou par un ou plusieurs textes informels. Si la question ou l'une des r´eponses est informelle, toute la d´ecision est informelle. Des r´eponses diff´erentes sont s´epar´ees par des virgules. Les valeurs sont repr´esent´ees par des expressions constantes, par des expressions constantes ayant un op´erateur comme pr´efixe ou par des gammes dont les limites sup´erieures et inf´erieures sont des expressions constantes. Les valeurs de r´eponse doivent | tre du m | me type que l'expression contenue dans la question.

Il est possible d'indiquer explicitement certaines r´eponses et de grouper toutes les autres r´eponses possibles en employant le mot-cl´e ELSE (AUTRE).

En LDS/PR, la d´ecision est repr´esent´ee par le mot-cl´e DÉCISION suivi par la sp´ecification de la question et la liste des r´eponses possibles, chacune ´etant associ´ee à la transition correspondante. Les r´eponses sont indiqu´ees entre parenth`eses. L'ensemble des transitions sortantes est d´elimit´e par le mot-cl´e ENDDECISION (FIN DE DÉCISION) plac´e 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 d´ecisions 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-cl´e ENDECISION (FIN DE DÉCISION). Les d´ecisions qui ne se terminent pas par un ´enonc´e terminal (c'est-`a-dire jonction, ´etat suivant, arr | t) continuent dans l'´enonc´e qui suit le mot ENDECISION, comme indiqu´e dans les deux branches ´equivalentes de la figure D-3.8.45.

Figure D-3.8.45, p. 35

L'instruction de d´ecision 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'À) comme en programmation structur´ee.

En LDS/GR, une d´ecision est repr´esent´ee à l'aide d'un symbole de d´ecision contenant le texte de la question. Le symbole doit avoir deux branches ou plus associ´ees aux r´eponses correspondantes. Chaque r´eponse doit | tre plac´ee 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 parenth`eses servant à d´elimiter les r´eponses sont facultatives mais il est sugg´er´e de les utiliser pour ´eviter tout malentendu.

On trouvera certains exemples de d´ecisions 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 appropriée 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 d´efinition de proc´edure ne peut exister que dans une d´efinition de processus, dans une d´efinition de service ou une d´efinition de proc´edure et, par cons´equent, une proc´edure n'est visible que pour le processus ou la proc´edure dans lesquels elle est d´efinie.

Une d´efinition de proc´edure se compose des ´el´ements suivants (dont certains sont facultatifs):

— nom de la proc´edure;

— param`etres formels de proc´edure: liste de noms de variables associ´ees à leur sorte. Ces param`etres servent à transf´erer l'information de la proc´edure ou à partir de celle-ci au moment de l'appel. Les param`etres de proc´edure peuvent | tre pass´es par valeur (param`etre IN) ou par r´ef´erence (param`etre IN/OUT). Si un param`etre est pass´e par valeur, la sp´ecification du param`etre formel d´efinit une variable locale à la proc´edure; s'il est pass´e par r´ef´erence, la sp´ecification d´efinit un synonyme pour la variable;

— d´efinitions de proc´edure: proc´edures qui peuvent | tre appel´ees tout comme la proc´edure proprement dite;

— d´efinitions de donn´ees: sp´ecification de types de donn´ees locales à la proc´edure;

— d´efinitions de variables: variables locales à la proc´edure;

— corps de proc´edure: sp´ecification du comportement effectif de la proc´edure pour ce qui est des ´etats et des actions (comme pour le corps de processus).

On trouvera dans la figure D-3.9.1 un exemple partiel de d´efinition de proc´edure en LDS/PR (les mots-cl´es du langage sont ´ecrits en majuscules). A noter que les param`etres 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 proc´edure*

Le corps de proc´edure pr´esente de grandes similitudes avec le corps de processus; toutefois les diff´erences sont les suivantes:

— la proc´edure termine son interpr´etation avec une indication <<retour>> au lieu d'une indication <<arr | t>>. En LDS/PR, l'´enonc´e de retour est repr´esent´e par le mot-cl´e RETURN;

— en LDS/GR, le symbole de d´ebut de proc´edure est l´eg`erement diff´erent du symbole de d´ebut de processus.

Les symboles de d´ebut et de retour de proc´edure sont indiqu´es dans le r´esum´e du LDS/GR.

Une proc´edure peut utiliser la construction avec branchements mais seulement pour se r´ef´erer à une ´etiquette interne. Un branchement peut ne pas | tre utilis´e ou | tre utilis´e pour acc´eder à une proc´edure de l'ext´erieur ou quitter celle-ci.

En LDS/GR, une d´efinition de proc´edure est repr´esent´ee par un diagramme de proc´edure tr`es similaire au diagramme de processus. Le diagramme de proc´edure comprend les ´el´ements suivants:

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

— l'en-t | te de proc´edure: le mot-cl´e PROCEDURE suivi du nom de la proc´edure et de la sp´ecification des param`etres formels de proc´edure. G´en´eralement, l'en-t | te de proc´edure est plac´e dans l'angle sup´erieur gauche du cadre ou, s'il n'y a pas de cadre, dans l'angle sup´erieur gauche du support sur lequel le diagramme est trac´e;

— une num´erotation de pages facultatives (a placer dans l'angle sup´erieur droit);

— des symboles de texte: dans le cas d'un diagramme de proc´edure, un symbole de texte peut | tre utilis´e pour contenir la sp´ecification de d´efinition de param`etres formels, de donn´ees et de variables;

— r´ef´erences de proc´edure: symboles de proc´edure, contenant chacun un nom de proc´edure repr´esentant une proc´edure locale d´efinie s´epar´ement;

— 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 proc´edure*

Les appels de proc´edure peuvent se produire chaque fois qu'une t | che et autoris´ee dans un graphe de processus ou de proc´edure. En un sens, une proc´edure peut | tre interpr´et´ee comme une t | che, avec les exceptions suivantes:

1) une proc´edure peut contenir des ´etats et, si tel est le cas, recevoir des signaux;

2) une proc´edure peut ´emettre des signaux. L'instance de processus d'origine est celle qui a appel´e la proc´edure.

Lorsqu'une proc´edure est appel´ee, son environnement est cr´e´e et l'interpr´etation de la proc´edure commence. Elle se poursuit jusqu'`a ce que l'on ait atteint l'indication RETURN (retour). Pendant l'interpr´etation de la proc´edure, tous les signaux adress´es au processus sont soit mis en r´eserve implicitement soit trait´es explicitement par la proc´edure. La proc´edure n'a pas sa propre file d'attente mais utilise celle du processus qui l'a appel´ee.

En LDS/PR, un appel de proc´edure est repr´esent´e par le mot-cl´e CALL suivi de l'identificateur de proc´edure et de la liste de param`etres r´eels mise entre parenth`eses. Si un param`etre n'est pas donn´e, il convient de l'indiquer par deux virgules cons´ecutives. Dans ce cas, le param`etre formel correspondant a la valeur <<ind´efinie>>. A noter aussi que la d´eclaration de IN ou IN/OUT est faite dans la d´efinition de proc´edure, de sorte qu'elle ne doit pas | tre r´ep´et´ee par l'´enonc´e 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] (a traiter comme tableau MEP), p. 43

En LDS/GR, un appel de proc´edure est repr´esent´e à l'aide d'un symbole d'appel de proc´edure contenant le nom de proc´edure et la liste des param`etres r´eels mise entre paranth`eses. On trouvera un exemple d'appel en LDS/GR dans la figure D-3.9.2.

Blanc