



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Z.333

LANGAGE HOMME-MACHINE

**MÉTHODOLOGIE POUR LA SPÉCIFICATION
DE L'INTERFACE HOMME-MACHINE**

OUTILS ET MÉTHODES

Recommandation UIT-T Z.333

(Extrait du *Livre Bleu*)

NOTES

1 La Recommandation Z.333 de l'UIT-T a été publiée dans le fascicule X.7 du Livre Bleu. Ce fichier est un extrait du Livre Bleu. La présentation peut en être légèrement différente, mais le contenu est identique à celui du Livre Bleu et les conditions en matière de droits d'auteur restent inchangées (voir plus loin).

2 Dans la présente Recommandation, le terme «Administration» désigne indifféremment une administration de télécommunication ou une exploitation reconnue.

© UIT 1988, 1993

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

Recommandation Z.333

MÉTHODOLOGIE POUR LA SPÉCIFICATION DE L'INTERFACE HOMME-MACHINE OUTILS ET MÉTHODES

1 Introduction

La présente Recommandation présente les outils et méthodes qui sont à la base de la procédure générale de travail décrite dans la Recommandation Z.332. Prises ensemble, les Recommandations Z.332 et Z.333 constituent la méthodologie pour la spécification de l'interface homme-machine.

2 Liste des outils et des méthodes¹⁾

Les outils et méthodes suivants sont nécessaires à l'application de la méthodologie pour la spécification des fonctions LHM:

- lignes directrices,
- modélisation,
- méthode de subdivision des fonctions LHM,
- métalangage de structure de l'information,
- méthode de description de procédure,
- représentation formelle de la structure syntaxique de chaque entrée et de chaque sortie.

3 Description des outils disponibles

3.1 *Lignes directrices*

3.1.1 *Pour la phase 1*

Déterminer pour chaque tâche:

- l'objet,
- les opérations que le système est supposé effectuer,
- les opérations que l'utilisateur est supposé effectuer,
- la complexité de la tâche du point de vue de l'utilisateur (voir la remarque),
- la fréquence de la tâche (voir la remarque),
- le niveau de la hiérarchie du réseau auquel la tâche est supposée être effectuée (central, centre d'exploitation et de maintenance),
- les aspects relatifs à la sécurité.

Remarque – On a admis ce qui suit pour mieux cerner ce que signifient la “fréquence” et la “complexité” d'une tâche.

3.1.1.1 *Fréquence*

Faible:

- si la tâche est supposée être effectuée à intervalles d'une semaine ou plus.

Moyenne:

- si la tâche est supposée être effectuée quotidiennement.

¹⁾ On peut, compte tenu de l'expérience des usagers, améliorer les outils et les méthodes, ce qui conduit à des adjonctions ou à des révisions.

Elevée:

- si la tâche est supposée être effectuée plusieurs fois par jour.

3.1.1.2 Complexité

Faible:

- petit nombre de paramètres (au sens général) – max. 0 : 3;
- la plupart de l'information associée à ces paramètres n'est pas composite;
- il n'y a pas de relation sémantique entre les différents paramètres et les valeurs de paramètre.

Moyenne:

- plus de 4 paramètres mais moins de 6 à 8;
- la plupart de l'information associée à ces paramètres est composite;
- il n'y a pas de relation sémantique entre les paramètres et/ou les valeurs de paramètre.

Grande:

- grand nombre de paramètres;
- la plupart de l'information associée à ces paramètres est composite;
- il y a des relations sémantiques entre les paramètres et/ou les valeurs de paramètre.

3.1.2 Pour la phase 2

Aucune ligne directrice spécifique n'est fournie pour la phase 2.

3.1.3 Pour la phase 3

On peut définir trois catégories principales de sortie dans la spécification de la sémantique des fonctions LHM, à savoir:

- 1) sorties de réponses dans le dialogue à destination des entrées de l'opérateur;
- 2) sorties de résultats dont l'utilisateur est censé être l'opérateur (par exemple, résultats des fonctions de rapport ou d'interrogation).
- 3) sorties de résultats dont l'utilisateur final n'est pas censé être l'opérateur (par exemple, données rassemblées en vue d'une mise au point ultérieure).

Le découpage des supports de sortie à utiliser et de leurs entités d'information subsidiaires ne doit pas être réalisé en détail, avec les lignes directrices suivantes:

- les supports et les caractéristiques de sortie qui admettront la première catégorie de sortie (sortie dans le dialogue) n'apparaîtront pas dans les diagrammes.
- les supports et les caractéristiques de sortie qui admettront la deuxième catégorie seront représentés comme l'indique la figure 1/Z.333.

Il est également reconnu que le niveau inférieur de détail, dont la définition dépendra des besoins de telle ou telle Administration, pourra en général comprendre des informations représentées à la figure 2/Z.333.

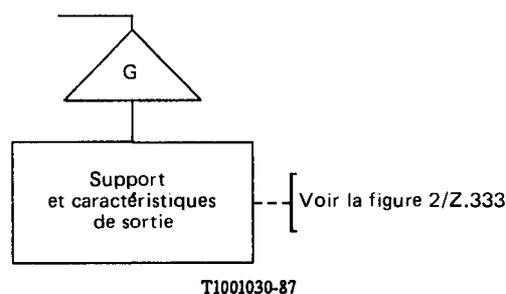


FIGURE 1/Z.333

**Support et caractéristiques de sortie
permettant d'admettre des sorties
dont l'utilisateur final est censé être l'opérateur**

Les supports de sortie qui admettront des sorties appartenant à la troisième catégorie seront représentés, si possible, de la même façon qu'au point précédent.

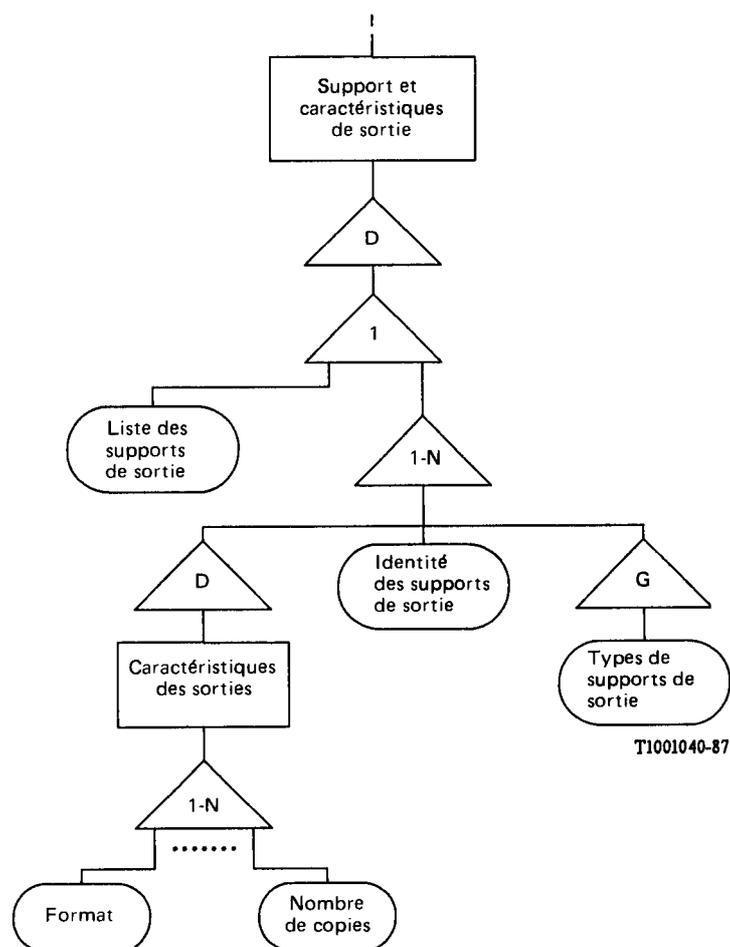


FIGURE 2/Z.333

Diagramme des supports de sortie et de leurs caractéristiques

3.1.4 Pour la phase 4

Pour définir les menus et formats individuels, suivre les lignes directrices concernant la conception des menus et des formats définies dans la Recommandation Z.323.

Pour définir les entrées et sorties individuelles:

- 1) Déterminer les opérations que le système est supposé effectuer.
- 2) Choisir des options dans la structure d'information des fonctions.
- 3) Définir l'information qui sera représentée par le code de commande ou son équivalent.
- 4) Définir l'information qui sera représentée par les paramètres et, si besoin est, leur ordre d'apparition.
- 5) Pour chaque paramètre, identifier, s'il y a lieu:
 - la gamme de valeurs,
 - les valeurs par défaut,
 - l'information qui sera fournie automatiquement par le système.
- 6) Définir les sorties de réponse dans le dialogue, les sorties de demande d'interaction et, le cas échéant, les sorties hors dialogue après avoir déterminé les diverses séquences d'opérations et les réactions de l'utilisateur aux sorties.

- 7) Définir la structure syntaxique associée.
- 8) Choisir les termes et abréviations pour les entrées et les sorties.

3.1.5 *Pour la phase 5*

- 1) Définir une procédure opérationnelle préliminaire en termes fonctionnels.
- 2) Mettre au point les procédures opérationnelles.

3.1.6 *Lignes directrices*

- 1) Vérifier que les fonctions LHM supportent les tâches à effectuer.
- 2) Il faudra considérer:
 - les aspects économiques,
 - la répartition correcte des compétences,
 - la définition correcte de la responsabilité,
 - la formation de l'utilisateur.

3.1.7 *Directives concernant l'harmonisation de la terminologie pour les phases 1 à 3*

Pour harmoniser la terminologie:

- 1) Utiliser le vocabulaire existant du CCITT.
- 2) Choisir les termes appropriés figurant dans la terminologie fonctionnelle générale (appendice I).
- 3) Elaborer des termes spécifiques, avec leurs définitions, appropriés au domaine fonctionnel en cause, compte tenu des considérations suivantes:
 - usage commun,
 - spécificité,
 - possibilité de traduction.

3.2 *Modélisation*

La modélisation suppose l'utilisation de textes et/ou de figures descriptifs élaborés soit en appliquant des symboles et des règles formels (modèle formel) soit sans appliquer de telles règles (modèle informel).

3.2.1 *Utilité des modèles*

Un outil disponible consiste à construire des modèles informels des parties des systèmes de télécommunications pour lesquelles le LHM est choisi comme langage de commande. L'organisation de l'Administration peut aussi faire l'objet de modèle. Plusieurs modèles peuvent s'appliquer quand on définit une tâche ou une fonction LHM. L'utilisation de modèles présente les avantages suivants:

- 1) Les modèles constituent un moyen d'échange de descriptions fonctionnelles.
- 2) La validité de l'interface homme-machine calculée peut être démontrée de manière cohérente par rapport aux modèles pertinents.

3.2.2 *Interprétation des modèles*

Un modèle peut être défini comme une abstraction d'une réalité vue sous un certain angle.

Dans les Recommandations de la série Z.300, le point de vue considéré est celui des usagers, c'est-à-dire des responsables des spécifications dans les Administrations et des concepteurs.

Il faudrait donc interpréter les modèles comme spécifications de haut niveau. Par ailleurs, ils n'ont pas pour objet de représenter, suggérer ou impliquer une mise en œuvre particulière.

Ils visent seulement à offrir un aperçu général, sur le plan théorique, des informations qui s'appliquent essentiellement à la commande de chaque domaine fonctionnel particulier et des principales relations existant entre les diverses entités du point de vue de l'opérateur.

Les modèles produits expressément pour déterminer la structure de commande LHM sont interprétés uniquement dans cette optique. D'autres modèles peuvent se prêter à la production de séquences de messages de commande LHM. La Commission d'études XI du CCITT s'estime tenue de produire des modèles qui puissent être liés aux méthodes de détermination de la structure d'information des fonctions LHM.

3.3 *Subdivision des fonctions LHM*

Les fonctions LHM générales sont structurées en fonctions LHM subsidiaires. Des niveaux multiples de subdivision sont permis (pour les exemples, voir les annexes à cette Recommandation).

3.4 *Métalangage de structure de l'information*

Chaque fonction LHM identifiée au niveau le plus bas de la subdivision des fonctions LHM est structurée en composants d'information qui sont nécessaires à sa mise en œuvre. Une structuration vers le bas est effectuée et des niveaux multiples de subdivision sont permis. L'outil est le métalangage présenté ci-après.

Pour mieux comprendre la structuration de l'information, on peut considérer une fonction LHM comme une action exercée sur un ou des objets. Les composants d'information peuvent alors se rapporter soit à des objets soit à des actions.

Une action générale associée à une fonction LHM peut être subdivisée en actions subsidiaires et en modificateurs de ces actions. Il est possible qu'aucune subdivision n'ait lieu. Cependant, si une subdivision est nécessaire, il convient de noter que, en ce qui concerne les actions, le terme "subdivision" implique que l'on détermine *aussi bien* les actions subsidiaires que tous les qualificatifs éventuels (modificateurs, options, etc.) associés à l'action. Il ne s'agit pas là d'une véritable "subdivision".

3.4.1 *Métalangage de subdivision*

3.4.1.1 *Considérations générales*

La représentation de la structure d'information associée à une fonction LHM suppose la spécification de toutes les entités d'information nécessaires et de leurs corrélations.

On peut obtenir cette représentation méthodiquement en utilisant des diagrammes de structure de l'information tracés à l'aide du métalangage décrit ci-après. Ce métalangage consiste en une série de symboles et de dessins conventionnels.

Un diagramme représente la structure d'information orientée du haut vers le bas, en commençant par l'identification de la fonction LHM à structurer et en finissant par tous les composants de l'information jugés nécessaires dans l'interfonctionnement homme-machine pour cette fonction.

Le processus de subdivision est accompli à l'aide de *séquences*, de *sélections* et d'*itérations*, qui permettent d'obtenir n'importe quel type de structure.

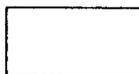
Sauf indication contraire, la séquence d'informations n'est pas implicite dans l'ordre de succession des différents éléments présentés dans les diagrammes.

3.4.1.2 *Entités d'information*

3.4.1.2.1 *Parties composites*

Une partie composite est une entité d'information qui peut être constituée d'éléments plus petits.

On utilise le symbole suivant:



3.4.1.2.2 Composant

Un composant est une entité d'information qui n'est pas intégrée dans une structure plus élaborée.

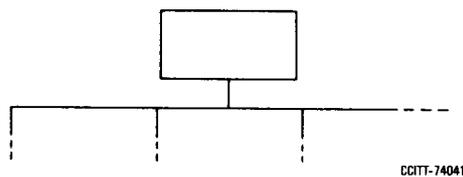
On utilise le symbole suivant:



3.4.1.3 Structuration

3.4.1.3.1 Subdivision

La subdivision dans les diagrammes de structure de l'information est représentée comme suit:



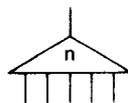
3.4.1.3.2 Séquence

Quand les entités d'information doivent être indiquées dans un ordre donné, elles sont spécifiées sous la forme d'une séquence. Ci-dessous, des flèches indiquent une séquence qui va de gauche à droite:

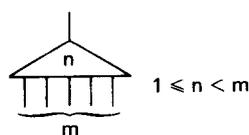


3.4.1.3.3 Sélection

Quand une partie composite comporte un certain nombre d'entités d'information, et lorsque quelques-unes ou une seulement de ces parties s'appliquent dans un cas donné, on a recours à un mécanisme de sélection, qui est représenté de la manière suivante:



Dans le cas général de sélection, il existe m possibilités de sélection, parmi lesquelles doit être choisi un nombre donné de possibilités n , de sorte que $n < m$.



CCITT-74041

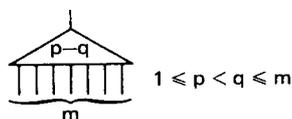
Le nombre n de possibilités à choisir est indiqué explicitement dans le symbole de sélection, tandis que le nombre total de possibilités correspond implicitement au nombre de points de sortie du symbole de sélection.

Les cas suivants sont autorisés:

$n = 1, m > 1$ Cas de sélection le plus courant qui implique le choix d'une des possibilités, et d'une seule.

$n > 1, m > n$ Sélection multiple de n possibilités parmi m possibilités.

Si le nombre des choix à faire peut varier entre une limite inférieure et une limite supérieure spécifiées, cela signifie un certain nombre d'options possibles. En pareil cas, les deux limites sont indiquées dans le symbole de sélection:



La limite inférieure p indique le plus petit nombre, et q indique le plus grand nombre de choix *différents* parmi les m possibilités. Il convient de noter que chaque choix ne peut être fait qu'une seule fois.

3.4.1.3.4 Options

Dans certains cas, des options peuvent être nécessaires, par exemple, des options de défaut, ou des options générales.

Le type d'option est alors indiqué par la lettre majuscule appropriée figurant seulement dans le symbole de sélection, à savoir D pour les options de défaut, et G pour les options générales. Un seul point de sortie du symbole est autorisé.

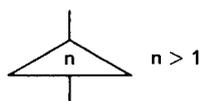


L'utilisation de l'option de défaut signifie que la valeur prise par une entité d'information sera automatiquement fournie si l'utilisateur ne donne pas une valeur à l'entrée.

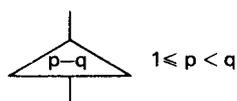
Une option générale est à utiliser pour diverses raisons reflétant les besoins des constructeurs ou des Administrations. Les entités d'information qui peuvent être déduites du point de sortie de cette case sont susceptibles, à titre facultatif, d'être intégrées dans l'interfonctionnement homme-machine. Cela signifie que ces informations existent dans le système d'une manière prédéterminée ou qu'elles ne sont pas nécessaires. Si cette distinction doit être faite, il faut rédiger une annotation renvoyant aux diagrammes de structure de l'information.

3.4.1.3.5 Itération

On utilise un mécanisme d'itération lorsqu'une partie composite est constituée d'entités d'information qui peuvent être répétées un nombre de fois arbitrairement choisi. L'itération est représentée par le symbole général suivant qui n'a qu'un point de sortie:



Si le nombre d'itérations peut varier entre certaines limites, le nombre minimum et le nombre maximum de répétitions d'une partie sont indiqués respectivement par p et q .



3.4.1.4 Conventions de tracé

3.4.1.4.1 Lignes de liaison et connecteurs

Chaque symbole est relié au symbole précédent par une ligne de liaison en trait plein.

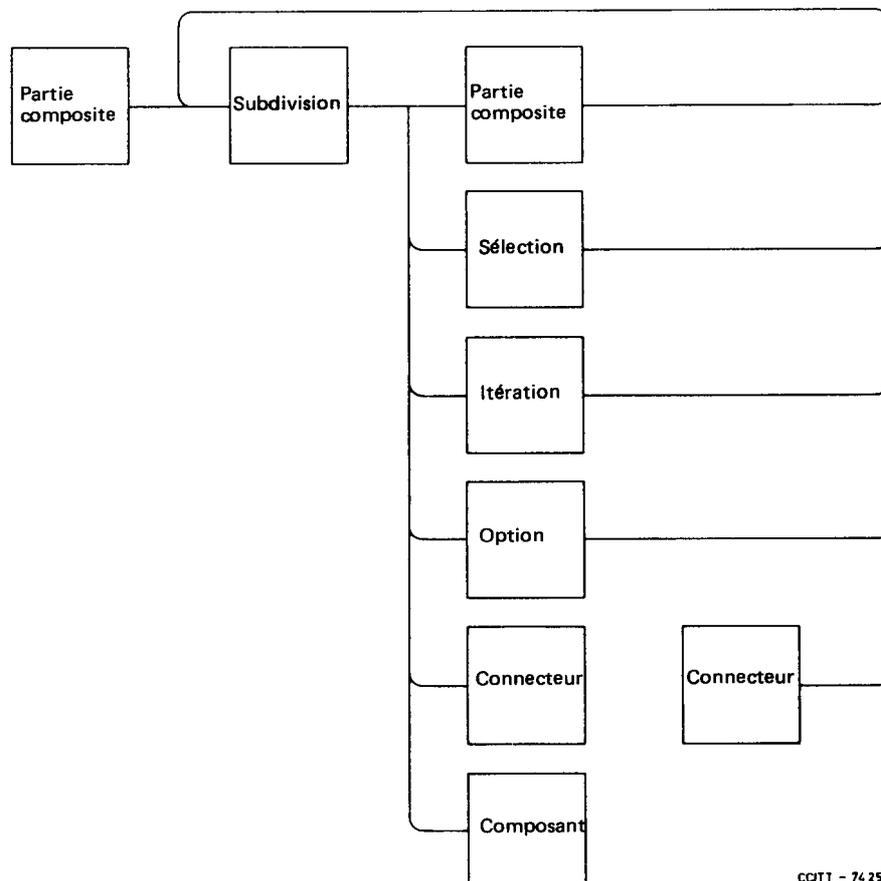
Une ligne de liaison en trait plein peut être interrompue par une paire de connecteurs associés, on suppose dans ce cas que la liaison s'effectue dans le sens allant du connecteur de sortie vers le connecteur d'entrée correspondant. Plusieurs connecteurs de sortie peuvent être associés au même connecteur d'entrée.

Le croisement des lignes de liaison doit être évité autant que possible.



3.4.1.4.2 Règles de connectivité

Chaque diagramme de structure de l'information commence par un symbole de partie composite et chaque ligne d'un diagramme se termine par un symbole de composant. Le tracé des diagrammes doit suivre les règles représentées ci-dessous pour les lignes de liaison.



Remarque 1 – Les types de symboles et la subdivision possible des lignes de liaison sont indiqués dans les cases carrées.

Remarque 2 – La subdivision comprend le cas banal d'une ligne de liaison continue unique.

3.4.1.4.3 *Annotations*

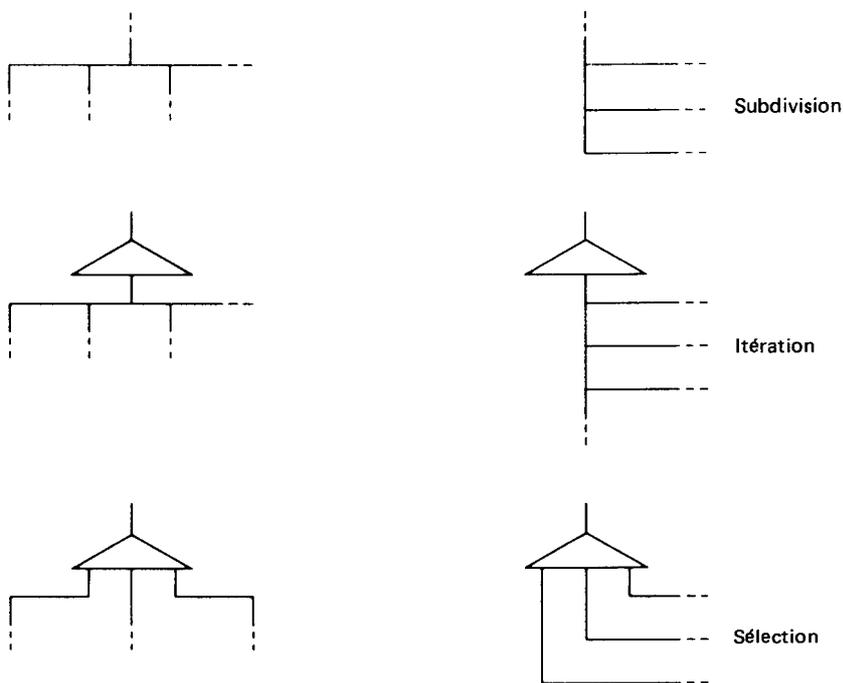
Les annotations sont désignées par le symbole suivant, où *n* est un nombre faisant référence à une note donnant une information descriptive ou explicative.

Annotation ----- [n]

Les annotations peuvent être reliées par une ligne tiretée à un symbole ou à une ligne de liaison quelconque.

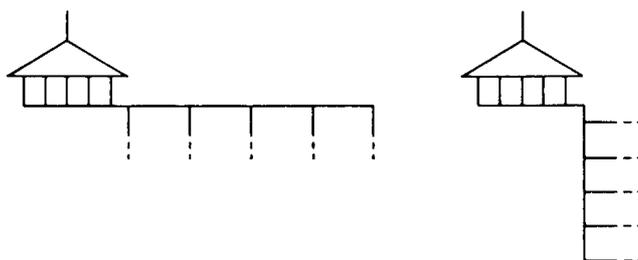
3.4.1.4.4 *Notations spéciales*

On peut utiliser des symboles verticaux au lieu des symboles horizontaux qui représentent normalement la structure, lorsque cela permet une présentation plus compacte. Les symboles verticaux peuvent être utilisés avec tous les types de structure.



CCITT-74280

Pour le symbole de sélection, dans le cas d'un nombre élevé de possibilités, les conventions de tracé suivantes sont également admises:



Quand le nombre d'entités d'information d'une structure est indéterminé, la représentation correspondante pourrait être la suivante:



CCITT - 74 260

Selon le type de structure utilisé.

3.5 *Méthode de description de procédures*

Le dialogue homme-machine peut être considéré comme une caractéristique d'un système SPC et peut être représenté au moyen de deux procédés: l'un lié à l'opérateur, l'autre lié à la machine. Ces deux procédés échangent des informations au moyen de signaux qui, aux fins du LHM, sont censés être surtout des entrées et des sorties.

Plus particulièrement, la description des procédures d'exploitation du LHM peut s'effectuer en attirant l'attention sur l'une des fonctions machine "logique", la fonction LHM associée et en décrivant le processus qui accomplit cette fonction.

Pour simplifier les dessins, il semble utile de limiter la description des signaux principaux entre l'opérateur et le système (entrées et sorties) et de ne pas représenter les caractéristiques telles que la base de temps, la signalisation des erreurs, les procédures d'édition, qui peuvent être décrites ailleurs au moyen du LDS, si besoin est (un exemple est fourni dans l'appendice II).

3.5.1 *Caractéristiques à utiliser dans la description*

Une procédure d'exploitation LHM peut être considérée comme un processus dont le comportement peut être spécifié en termes d'entrées, d'états, de transitions, de décisions, de sorties et de tâches.

Dans les paragraphes qui suivent, les notions fondamentales du LDS sont interprétées dans le cadre du LHM.

3.5.1.1 *Entrée*

Une entrée est une série de données introduite par l'utilisateur et reconnue par la procédure d'exploitation LHM. L'entrée peut être constituée, par exemple, de commandes en introduction directe d'information, ou d'autres types de données.

3.5.1.2 *Etat*

Un état est une condition dans laquelle l'action de la procédure LHM est suspendue en attendant une entrée.

3.5.1.3 *Transition*

Une transition est une séquence d'actions qui se produit lorsqu'une procédure d'exploitation LHM passe d'un état à un autre en réaction à une entrée.

3.5.1.4 *Décision*

Une décision est une action dans une transition qui pose une question à laquelle une réponse peut être obtenue au même instant et choisit l'un des nombreux trajets possibles pour continuer les transitions.

3.5.1.5 *Sortie*

La sortie est une série de données obtenue par la procédure d'exploitation LHM et qui agit à son tour comme une entrée dans le processus d'exploitation.

3.5.1.6 *Tâche*

Une tâche est une action, au cours d'une transition, qui n'est ni une décision, ni une sortie.

3.5.1.7 *Symboles et règles*

Les symboles et règles sont ceux qui sont définis dans les Recommandations LDS de la série Z.100.

3.6 Représentation formelle de la structure syntaxique des entrées et des sorties spécifiques

La représentation formelle de la structure syntaxique des entrées et des sorties spécifiques peut être fournie grâce au métalangage syntaxique de la Recommandation Z.302. L'utilisation de la Forme de Backus Naur (FBN) a aussi été proposée, car elle pourrait être plus efficace. Il peut être nécessaire de faire des adjonctions à mesure que des possibilités de terminaux évolués sont considérées par le sous-groupe LHM. Il convient d'étudier l'utilité de ces méthodes et, si possible, d'en recommander une seule.

3.6.1 Forme de Backus Naur (FBN)

Les entrées et les sorties sont définies comme des séquences d'éléments terminaux et/ou non terminaux.

Les éléments terminaux sont des caractères appartenant au jeu de caractères LHM et définis dans la Recommandation Z.314; les éléments syntaxiques correspondent aux définitions des Recommandations Z.314, Z.315 et Z.316. Les éléments syntaxiques sont indiqués par leur nom écrit en minuscules entre des parenthèses coudées (< et >).

Les éléments non terminaux sont des éléments qu'il faudra à nouveau définir en tant que séquences d'éléments terminaux et/ou non terminaux. Ils sont indiqués par un ou plusieurs mots écrits en minuscules entre les signes (< et >).

3.6.1.1 Notation

Les définitions sont indiquées par des commandes ou des éléments non terminaux à gauche du symbole ::= (deux fois deux points, signe d'égalité) et, à droite, par une ou plusieurs séquences d'éléments terminaux et/ou non terminaux.

Les solutions de rechange sont indiquées, précédées d'une barre verticale (|).

Les éléments terminaux et non terminaux peuvent être regroupés grâce à l'utilisation d'accolades ({ et }), la répétition de ces groupes est indiquée au moyen de deux indices après les accolades, un pour le nombre minimal, un pour le nombre maximal de répétitions du groupe.

Si un groupe d'éléments terminaux et non terminaux est placé entre crochets ([et]), il est facultatif.

Un exemple est fourni dans l'appendice III.

APPENDICE I

(à la Recommandation Z.333)

Glossaire de termes couramment utilisés dans la spécification de l'interface homme-machine

Ce glossaire de termes courants doit être utilisé, s'il y a lieu, par les organes du CCITT, dans l'application des phases 1 à 3 de la méthodologie. Il devrait pouvoir être complété à mesure que la méthodologie est appliquée à une gamme plus étendue de domaines. Le présent document n'a pour objet de restreindre le choix de termes dont disposent les constructeurs et les Administrations pour représenter ces concepts à l'interface homme-machine.

Il a été indiqué dans la Recommandation Z.332 qu'il est utile de considérer les fonctions du LHM en tant qu'*actions* exercées sur des *objets*. Les concepts représentés par les termes sélectionnés ici sont limités à ceux qui se rapportent aux actions. On prévoit qu'à mesure que ce glossaire sera élargi, la plupart des concepts relatifs aux actions seront définis ici car ils sont généralement utilisés dans d'autres domaines fonctionnels. Inversement, les concepts concernant les objets seront en général spécifiques à un domaine fonctionnel et sont donc définis, dans le glossaire, conjointement avec un domaine fonctionnel.

Parmi les concepts relatifs à des actions pouvant être exécutées à l'interface homme-machine, il en est dont l'objet propre de l'action concerné:

- les données seulement,
- l'équipement seulement,
- les données ou l'équipement.

Ces trois catégories d'action correspondent aux trois grandes divisions de ce glossaire.

Un certain nombre de concepts indiqués ci-après se comprennent mieux lorsqu'ils sont présentés en paires complémentaires et s'utilisent généralement ainsi d'ailleurs; ces cas seront indiqués par la notation suivante: par exemple, CRÉATION/SUPPRESSION.

I.1 *Actions de gestion des données*

Le terme "**ensemble de données**" est défini comme ensemble accessible par l'utilisateur d'un ou plusieurs éléments d'information caractérisés par un usage particulier et également par les contraintes imposées à la présentation des données et/ou aux valeurs qui le rendent appropriés à cette utilisation.

I.1.1 *CRÉATION/SUPPRESSION*

Les concepts suivants concernent la commande par l'utilisateur de l'existence d'ensembles de données dans le système.

CRÉATION: Etablir, dans le système, un nouvel ensemble de données.
Exemples: CRÉATION D'UN ENSEMBLE DE MESURES, CRÉATION D'UNE LISTE D'OBJETS.

SUPPRESSION: Eliminer un ensemble de données du système.
Exemples: SUPPRESSION D'UN ENSEMBLE DE MESURES, SUPPRESSION D'UNE LISTE D'OBJETS.

I.1.2 *MODIFICATION ET ÉDITION*

La modification des données est généralement accomplie en appliquant l'une des deux méthodes fondamentales indiquées ci-après. La première méthode (MODIFICATION) consiste à utiliser des entrées et sorties fonctionnelles spécifiques visant à modifier des types particuliers d'ensembles de données ou même des éléments d'information particuliers dans ces ensembles de données. La seconde méthode de modification des données (ÉDITION) permet à l'utilisateur d'apporter directement des modifications sur un écran où sont affichées les données à modifier.

Compte tenu de cela, les organes du CCITT qui appliquent la méthodologie décrite dans la présente Recommandation devraient employer le terme MODIFICATION pour toute modification devant être apportée aux données, sauf dans les cas où la possibilité d'ÉDITION présenterait des avantages évidents, comme dans l'exemple indiqué ci-après.

MODIFICATION: Modifier des éléments d'information spécifiés dans un ensemble de données par l'intermédiaire d'une entrée ou d'entrées conçues à cet effet.
Exemple: MODIFICATION DES SEUILS D'ANALYSE.

ÉDITION: Visualiser un ensemble spécifié de données et modifier ensuite l'ensemble de données. Une possibilité commune du système, par exemple, le programme d'"édition", est généralement utilisée comme support d'une telle action.
Exemple: ÉDITION D'ENREGISTREMENT DE DONNÉES CONCERNANT LE TRAFIC.

I.1.3 *ACTIVATION/DÉSACTIVATION*

La création d'un ensemble de données ne signifie pas nécessairement que les données sont immédiatement mises à disposition par le système pour son application envisagée. Les concepts suivants ont pour objet de rendre disponible ou non au système un ensemble de données déjà créé.

ACTIVATION: Déclencher une procédure de système qui exige l'introduction préliminaire des données ou mettre à la disposition du système, pour son application envisagée, un ensemble de données déjà introduit.
Exemples: ACTIVATION D'UNE MESURE, ACTIVATION D'UN ESSAI PÉRIODIQUE.

DÉSACTIVATION: Mettre fin à un processus de système déclenché par une action d'ACTIVATION ou faire en sorte qu'un ensemble de données ne soit pas disponible pour utilisation par le système.

Exemples: DÉSACTIVER UNE MESURE, DÉSACTIVER UN ESSAI PÉRIODIQUE.

I.1.4 *FILTRE ET TRI*

Ces concepts permettent à l'utilisateur de manipuler des données qui seront stockées ultérieurement ou auxquelles on pourra avoir accès.

FILTRE: Former un sous-ensemble de l'ensemble de données comprenant tous les éléments d'information de l'ensemble répondant aux critères spécifiés. L'ensemble initial de données n'est pas affecté par cette action.

Exemple: RAPPORTS SUR DES PROBLÈMES CONCERNANT LE FILTRE OU RAPPORT DE REMISE EN ACTIVITÉ.

TRI: Remanier l'ordre de présentation d'un ensemble de données d'après des critères spécifiés (ou valeur par défaut). Cette action n'affecte pas le contenu de l'ensemble initial mais uniquement l'ordre de présentation.

Exemple: TRI D'UN FICHIER DE NOMS (par exemple, suivant l'ordre alphabétique).

I.1.5 *INTERROGATION ET LECTURE*

Les concepts ci-après décrivent des actions exécutées par le système qui permettent à l'utilisateur d'avoir accès à des parties déterminées des données créées par l'utilisateur ou par le système.

INTERROGATION: Permet une visualisation des valeurs actuelles des informations dans un ensemble de données ou plus.

Exemples: INTERROGATION D'UNE MESURE, INTERROGATION D'UN TYPE DE MESURE.

LECTURE: Visualiser successivement les valeurs actuelles des informations dans un ensemble de données. L'utilisateur peut examiner les éléments d'information vers l'avant ou vers l'arrière.

Exemple: LECTURE DES FICHIERS DE RAPPORTS.

I.1.6 *ENTRÉE/SORTIE ET ACHEMINEMENT*

Les concepts définis dans cette section concernent le transfert des données d'un emplacement à un autre.

ENTRÉE: Introduction de données dans le système à l'aide d'un terminal d'utilisateur.

Exemple: RAPPORT SUR DES PROBLÈMES CONCERNANT L'ENTRÉE OU RAPPORT DE REMISE EN ACTIVITÉ.

SORTIE: Transférer des données précises du système au terminal d'utilisateur (par exemple, TDV, imprimante).

Exemple: RAPPORT RÉSUMÉ DE SORTIE.

La distinction entre la SORTIE et l'INTERROGATION (I.1.5) est la suivante: l'INTERROGATION donne simplement une relecture des données créées par l'utilisateur, tandis que la SORTIE se réfère aux données sur lesquelles le système lui-même a agi d'une certaine manière, par exemple, les rapports.

ACHEMINEMENT: Indiquer au système que tout message ultérieur, classe de données ou type de message indiqué doit être acheminé vers des supports spécifiés.

Exemple: ACHEMINEMENT DE LA SORTIE DE RAPPORTS.

I.2 *Actions concernant la gestion de l'équipement*

I.2.1 *RETRAIT/RÉTABLISSEMENT ET POSITIONNEMENT*

Il arrive souvent que des organes d'équipement puissent simplement être mis hors service ou en service sous commande par logiciel. La paire RETRAIT/RÉTABLISSEMENT représente ces deux actions. La manipulation du statut des objets avec un ensemble plus compliqué d'états de maintenance est exprimée par l'action de POSITIONNEMENT qui s'applique aussi en général aux états hors service et en service. La paire RETRAIT/RÉTABLISSEMENT est utilisée fréquemment et suffit à une gamme étendue d'équipements, aussi est-elle signalée ici comme cas spécial important de l'action de POSITIONNEMENT.

RETRAIT: Mettre hors service des organes d'équipement déterminés. Le système maintient un contact avec ces organes d'équipement, de sorte qu'ils peuvent être remis en service par l'action de RÉTABLISSEMENT définie ci-après, par reprise automatique ou dérogation manuelle.

Exemple: RETRAIT D'UN CIRCUIT.

RÉTABLISSEMENT: Remettre en service des organes d'équipement déterminés.

Exemple: RÉTABLISSEMENT D'UN CIRCUIT.

POSITIONNEMENT: Placer l'équipement dans un état déterminé (nombre d'états > 2). Parmi les états possibles: mise en service et mise hors service.

Exemple: POSITIONNEMENT D'UN ORGANE D'ÉQUIPEMENT.

I.2.2 *AUTORISATION/INTERDICTION*

Les systèmes modernes (par exemple, pour la maintenance ou la commande) emploient de nombreuses fonctions du système qui se produisent automatiquement ou dépendent uniquement de la détection de certaines conditions. Il est souvent essentiel de pouvoir donner l'instruction au système de *ne pas* exécuter ces fonctions, même si toutes les conditions voulues sont réunies. La possibilité complémentaire de remettre la fonction à commande automatique à son état normal doit alors être prévue.

AUTORISATION: Permettre que des actions, réponses ou fonctions déterminées du système se produisent. Ces fonctions peuvent être interdites par la conception du système ou par l'action d'INTERDICTION du système définie ci-après.

Exemple: AUTORISATION D'UN SEUIL.

INTERDICTION: Empêcher que des actions, réponses ou fonctions déterminées du système se produisent. Ces fonctions peuvent normalement être autorisées par la conception du système ou par l'action d'AUTORISATION définie ci-dessus.

Exemple: INTERDICTION D'UN SEUIL.

I.3 *Actions de gestion pouvant s'appliquer aux données ou à l'équipement*

INITIALISATION: Appliquer aux données ou à l'équipement spécifiés une condition ou une valeur initiale (normale) définie au préalable.

Exemples: INITIALISATION D'UN COMPTEUR DE SEUIL, INITIALISATION D'UN DISPOSITIF DE SORTIE.

EXÉCUTION: Exécuter une procédure définie au préalable.

VÉRIFICATION: Veiller à l'observation d'une règle de compatibilité sur un ensemble spécifié de données.

CONNEXION: Assurer une connexion entre deux entités existantes.

DÉCONNEXION: Interrompre une connexion déjà établie.

DÉBUT: Entamer une procédure ou un processus.

ARRÊT: Mettre fin à l'activité spécifiée et laisser le système dans un état défini.

SUSPENSION: Interrompre temporairement une activité.

REPRISE: Continuer une activité suspendue au préalable.

APPENDICE II

(à la Recommandation Z.333)

Exemple de la méthode de description de procédure

La tâche “créer une nouvelle mesure du trafic” est décrite comme une procédure dans laquelle deux processus LDS différents interviennent: le processus usagers et le processus système.

Dans les diagrammes suivants ne figurent que les aspects pertinents de la procédure; certaines opérations sont omises: sortie de rejet due à des erreurs syntaxiques et procédures de correction connexes, etc., qui sont communes aux autres procédures.

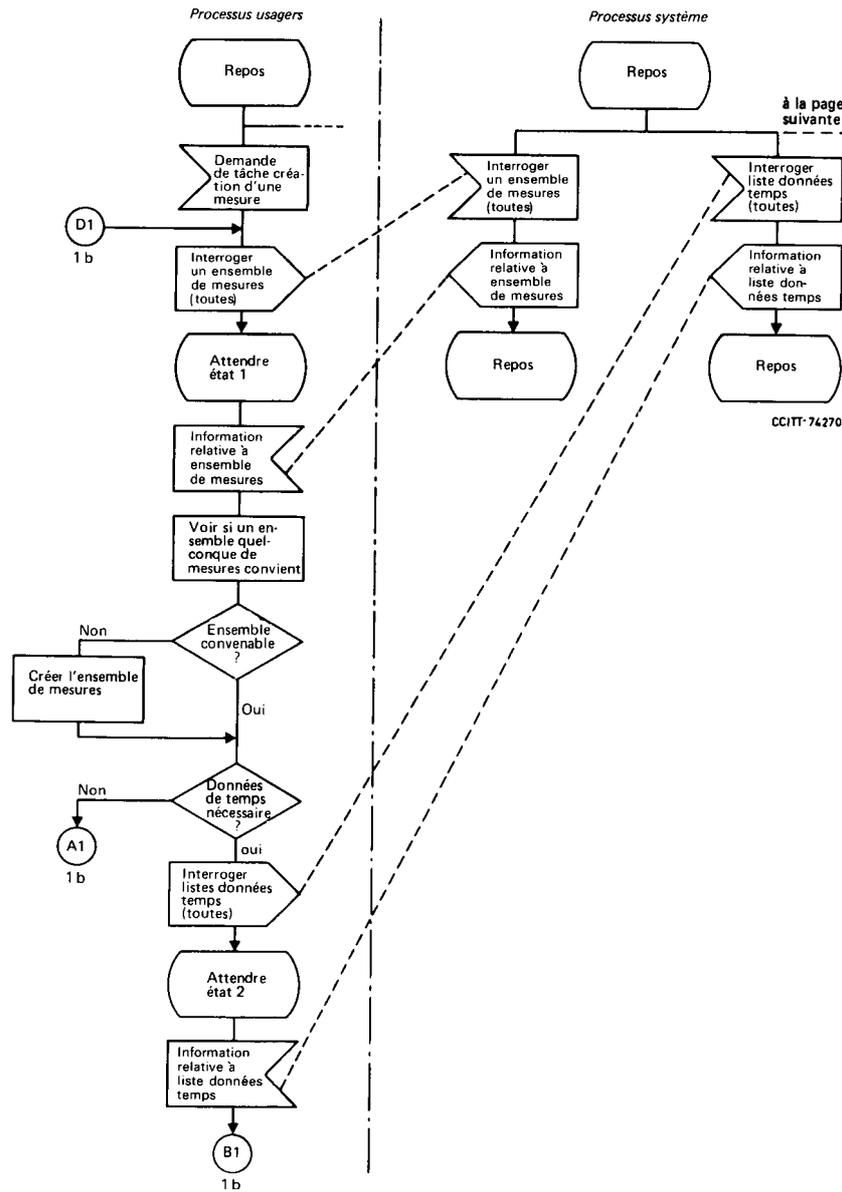


FIGURE II-1a/Z.333

Exemple de description de procédure

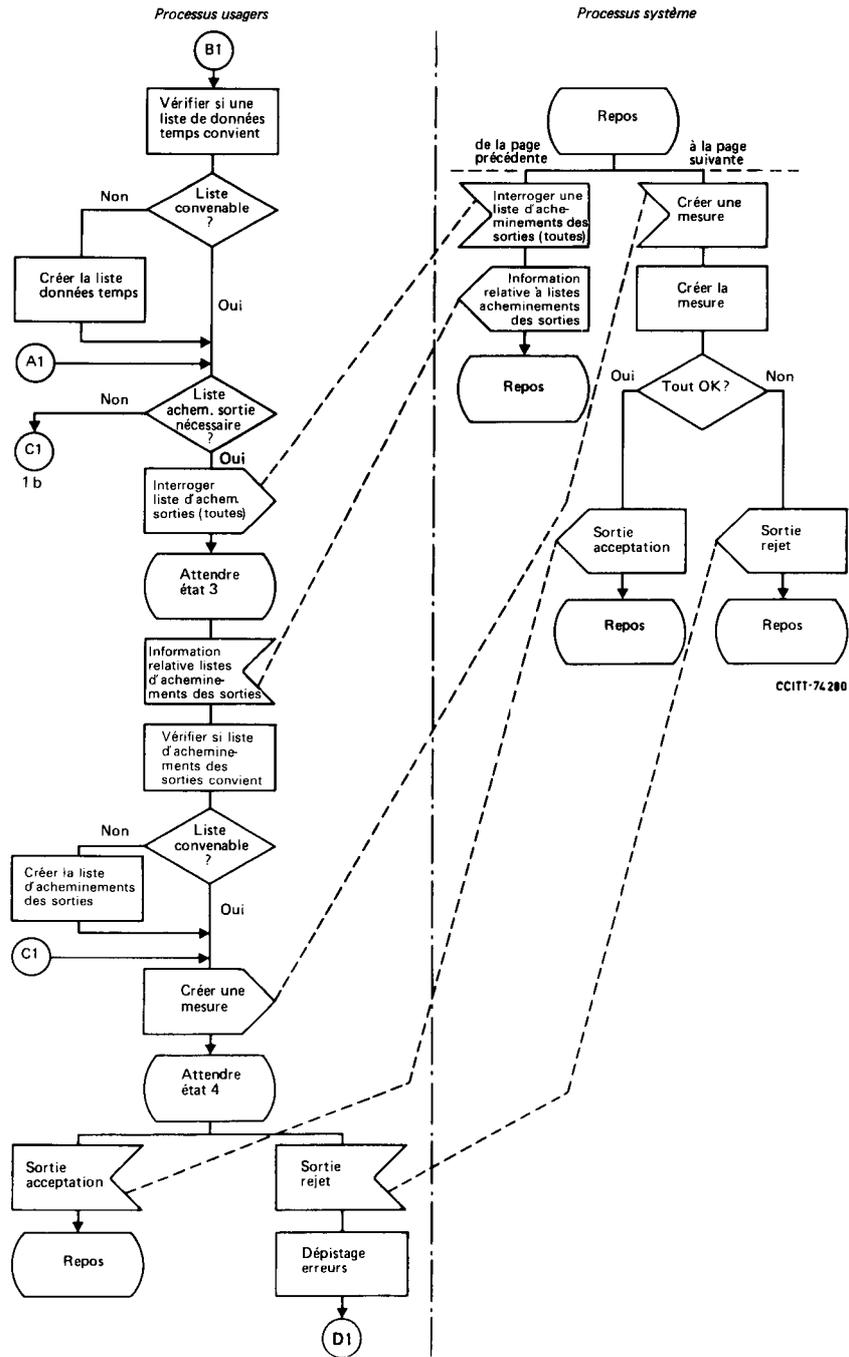


FIGURE II-1b/Z.333

Exemple de description de procédure (suite)

APPENDICE III

(à la Recommandation Z.333)

Exemples de l'utilisation de la Forme de Backus Naur (FBN)

En appliquant le métalangage FBN décrit au § 2.6.1 de l'exemple de fonctions de gestion de mesure du trafic (voir l'annexe A à la Recommandation Z.336) – figures B-9/Z.336 et B-14/Z.336), on obtient les exemples FBN suivants, dans l'hypothèse selon laquelle il existe une relation biunivoque entre la fonction LHM et la commande associée:

a) *Fonction "créer une liste d'objets":*

```
< créer une liste d'objets > ::= < code de commande >
                                < identité de la liste d'objets >
                                {,< liste d'objets d'un type >};
                                1 – N

< identité de la liste d'objets > ::= < nom du paramètre > = < nom symbolique >
< liste d'objets d'un type > ::= < type d'objet > = < identité des objets >
< type d'objet > ::= < nom du paramètre >
< identité de l'objet > ::= < nombre décimal > { {& < nombre décimal > } }
                                {&& < nombre décimal > } } |
                                O – N
                                < nom symbolique > {& < nom symbolique >}
                                O – N
```

b) *Fonction "supprimer une liste d'objets":*

```
< supprimer une liste d'objets > ::= < code de commande >
                                < liste d'identités d'une liste d'objets >;
< liste d'identités d'une liste d'objets > ::= < nom du paramètre > =
                                < nom symbolique > {& < nom symbolique >}
```