

N° d'ordre : 166

50376
1987
235

50376
1987
235

THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE FLANDRES ARTOIS

pour obtenir le titre de

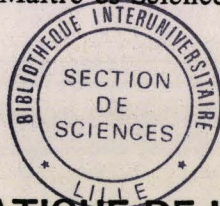
DOCTEUR

spécialité : Automatique

par

HE Bing

Maitre ès Sciences



**GENERATION AUTOMATIQUE DE LOGICIELS D'AIDE A LA
COMMUNICATION POUR HANDICAPES**



Soutenue le 20 octobre 1987 devant la Commission d'Examen

Messieurs

P. VIDAL
J.M. TOULOTTE
J. BREMONT
C. VASSEUR
C. LARROQUE
J. TICHON

Madame
Monsieur

Président
Directeur du travail
Rapporteur
Rapporteur
Invitée
Invité

AVANT-PROPOS

Le travail de recherche présenté dans ce mémoire a été effectué au Centre d'Automatique de l'Université des Sciences et Techniques de Lille Flandres Artois.

Avant d'exposer cette étude, je tiens à exprimer ma profonde gratitude à Monsieur le Professeur Pierre VIDAL, Directeur du Centre d'Automatique, pour l'accueil qu'il m'a réservé. Je suis très honoré qu'il ait accepté de présider le jury de cette thèse.

Que Monsieur le Professeur Jean-Marc TOULOTTE, Directeur de cette thèse, trouve ici l'expression de mes plus vifs remerciements pour les conseils qu'il m'a prodigués tout au long de ce travail, ainsi que la confiance témoignée.

Que Monsieur Jacques BREMONT, Professeur à l'Université de Nancy I soit remercié pour l'intérêt qu'il porte à ce travail et pour être rapporteur de cette thèse.

Que Monsieur Christian VASSEUR, Professeur à l'Université des Sciences et Techniques de Lille Flandres Artois trouve ici l'expression de mes plus vifs remerciements pour être rapporteur de cette thèse.

Je tiens à remercier Madame Christine LARROQUE, Orthophoniste à l'Hôpital des Invalides de Paris pour avoir accepté de juger ce travail.

Je tiens à remercier Monsieur Jacques TICHON, Enseignant à l'Institut Supérieur Industriel de Bruxelles et chercheur au Centre d'Automatique, pour les nombreuses discussions que nous avons eues et pour sa participation au Jury.

J'adresse mes vifs remerciements au personnel et aux collègues du Centre d'Automatique, qui m'ont accueilli, aidé et soutenu tout au long de ce travail, en particulier Monsieur Bernard CEURSTEMONT dont l'aide a été si souvent sollicitée, Madame Annick PIGNON pour ses nombreuses aides, et Monsieur HOUZE pour la réalisation matérielle de cette thèse.

TABLE DES MATIERES

<u>INTRODUCTION GENERALE</u>	1
CHAPITRE I :	
<u>GENERALITES SUR L'AIDE A LA COMMUNICATION POUR HANDICAPES</u>	
<u>Introduction</u>	3
<u>I.1 Besoins de systèmes d'aide à la communication pour handicapés</u>	3
<u>I.2 Systèmes d'aide à la communication pour handicapés</u>	4
I.2.1 Comportement extérieur d'un système d'aide à la communication	4
I.2.2 Fonction d'un système d'aide à la communication	5
I.2.3 Composants matériels d'un système d'aide à la communication	5
<u>I.3 Environnement de la communication</u>	8
<u>I.4 Eléments (ou constituants) primaires dans un système d'aide à la communication</u>	8
I.4.1 Eléments informatifs(pavés affichables)	8
I.4.2 Eléments de commande(pavés exécutables)	9
I.4.3 Organisation logique des constituants primaires	10
<u>I.5 Sélection des constituants primaires</u>	12
I.5.1 Interfaces(ou organes) d'accès adaptés	12
I.5.2 Méthodes de sélection des constituants	12
I.5.2.1 Sélection directe sans codage	13
I.5.2.2 Sélection avec codage	13
I.5.2.3 Sélection par balayage	16
<u>I.6 Mémorisation des messages primaires</u>	20
<u>I.7 Visualisation des informations dans l'aide à la communication</u>	22
I.7.1 Visualisation des informations	22
I.7.1.1 Visualisation des constituants primaires	22
I.7.1.2 Visualisation des messages primaires	22
I.7.1.3 Visualisation de renseignements divers	23
I.7.2 Ecrans de visualisation des informations	23
<u>I.8 Accélération à la construction des messages primaires</u>	23
I.8.1 Méthode d'accélération fréquentielle statique	25
I.8.2 Méthode d'accélération fréquentielle dynamique	25
I.8.3 Méthode d'accélération par expansion	26
I.8.4 Méthode d'accélération syntaxique	26
I.8.5 Méthode d'accélération sémantique	27
<u>I.9 Transformation des messages</u>	27
I.9.1 Messages primaires	27
I.9.2 Messages secondaires	27
I.9.3 Transformation des messages primaires en messages secondaires	27

I.10. <u>Emission des messages secondaires</u>	28
<u>Conclusion</u>	28

CHAPITRE II :

**CONCEPTION DE LA STRUCTURE D'UN SYSTEME GENERAL
INDIVIDUALISABLE D'AIDE A LA COMMUNICATION POUR
HANDICAPES**

<u>Introduction</u>	30
II.1 <u>Choix du matériel et du logiciel de base pour le système de communication</u>	30
II.1.1 Choix du matériel de base	30
II.1.2 Choix du langage de programmation	31
II.2 <u>Structuration des constituants primaires</u>	31
II.2.1 Structure du constituant primaire	31
II.2.2 Structure du menu	37
II.3 <u>Généralisation de la procédure principale du fonctionnement</u>	41
II.4 <u>Conception des méthodes de sélection</u>	41
II.4.1 Introduction	41
II.4.2 Sélection directe sans codage	43
II.4.3 Sélection avec codage ligne-colonne(ou colonne-ligne)	43
II.4.4 Sélection avec codage dichotomique mixte	45
II.4.5 Sélection par balayage pavé par pavé	45
II.4.6 Sélection par balayage ligne-colonne(ou colonne-ligne)	49
II.4.7 Sélection par balayage dichotomique mixte	49
II.5 <u>Conception de la mémorisation des messages primaires</u>	52
II.5.1 Structure du tampon de message	52
II.5.2 Mémorisation des messages primaires	54
II.5.3 conception des pavés vides	54
II.6 <u>Conception de la visualisation des messages primaires</u>	57
II.6.1 Multi-fenêtrage de l'écran vidéo	57
II.6.2 Structure du tampon d'affichage de message	57
II.6.3 Visualisation des messages primaires	60
II.7 <u>Intégration des transformations dans le système de communication</u>	60
II.8 <u>Intégration des méthodes d'accélération dans le système de communication</u>	63
II.8.1 Introduction	63
II.8.2 Structures des données associées à la procédure d'accélération	63
II.8.3 Procédure d'accélération	66
II.9 <u>Conception des éléments de commande</u>	66
II.9.1 Commandes de contrôle d'édition des messages	66
II.9.2 Commandes de contrôle de l'environnement	70
<u>Conclusion</u>	71

CHAPITRE III :

GENERATION AUTOMATIQUE DE LOGICIELS DE COMMUNICATION POUR HANDICAPES -- CONCEPTS GENERAUX

<u>Introduction</u>	72
<u>III.1 Production automatique de logiciels d'application</u>	72
III.1.1 Progiciels individualisables	72
III.1.2 Langages de très haut niveau	73
III.1.3 Programmation évolutive	73
III.1.4 Programmation par exemple	74
III.1.5 Systèmes à banque de connaissances	75
<u>III.2 Etude détaillée de l'approche "progiciels individualisables"</u>	76
III.2.1 Principe de génération	76
III.2.2 Phases de génération d'un produit logiciel	76
III.2.3 Mérites et limites de l'approche	77
<u>III.3 Justification du choix de l'approche</u>	78
<u>III.4 Progiciel individualisable d'aide à la communication pour handicapés</u>	79
III.4.1 Schéma de l'architecture et du fonctionnement	80
III.4.2 Spécification des besoins	82
III.4.2.1 Spécification via le questionnaire	82
III.4.2.2 Edition des données de base	83
III.4.2.3 Modèle utilisateur	84
III.4.3 Production du logiciel source	84
III.4.3.1 Ensemble de modules prédéfinis	85
III.4.3.2 Génération de la partie "programmes"	86
III.4.3.3 Génération de la partie "données"	87
III.4.4 Test du logiciel sur le système de génération	87
III.4.5 Production du logiciel objet	87
III.4.6 Documentation du logiciel généré	88
III.4.7 Archivage et réutilisation des spécifications	88
<u>Conclusion</u>	89

CHAPITRE IV :

IMPLEMENTATION DU SYSTEME DE GENERATION DE COMMUNICATEURS POUR HANDICAPES

<u>Introduction</u>	90
<u>IV.1 Configurations matérielles et logicielles</u>	90
IV.1.1 Configuration APPLE II -- HHC	90
IV.1.2 Configuration IBM PC	91
<u>IV.2 Architecture et principe de fonctionnement</u>	92
IV.2.1 Architecture du système	92
IV.2.2 Processus de génération	94
<u>IV.3 Editeur interactif de communicateur</u>	94
IV.3.1 Principe de l'édition de communicateur	94
IV.3.2 Modes de travail et commandes	95
IV.3.3 Interaction opérateur-éditeur	102

IV.3.4 Représentations des objets manipulés et structure du programme	104
IV.3.4.1 Représentation des objets manipulés	104
IV.3.4.2 Structure du programme	104
IV.4 <u>Générateur de communicateur</u>	105
IV.4.1 Génération des données	105
IV.4.2 Construction du programme	105
IV.4.3 Compilation et mise en forme	106
IV.5 <u>Tests et essais des communicateurs générés</u>	106
IV.6 <u>Exemples d'utilisation du système</u>	107
IV.6.1 Edition de pavés et menus	107
IV.6.2 Communication	114
<u>Conclusion</u>	118
<u>CONCLUSION GENERALE</u>	119
<u>BIBLIOGRAPHIE</u>	120
<u>ANNEXES</u>	
A.1. Le BLISS	131
A.2. Code Pictographique du GRACH	136

INTRODUCTION GENERALE

Le travail présenté dans ce mémoire est effectué au laboratoire du Centre d'Automatique de l'Université des Sciences et Techniques de Lille Flandre Artois, et il s'inscrit dans le cadre de la production automatique de logiciels d'application (CAO et PAO de logiciels) et au domaine de la rééducation des handicapés physiques. Il s'agit plus précisément d'appliquer des techniques ou approches informatiques à la conception et la réalisation de systèmes d'aide à la communication pour handicapés.

Pour la plupart des handicapés physiques, se pose le problème de la communication avec le monde extérieur. Pour les aider à communiquer, il y a grand intérêt à réaliser des systèmes informatiques d'aide à la communication, dits communicateurs.

Un système de communication, développé de façon individuelle pour un cas de handicap, est rarement adapté à un autre cas, et il est souvent onéreux vu l'effort et le temps importants nécessaires à son développement. Un moyen pour permettre de palier ces problèmes consiste à structurer sur le plan du matériel un système de communication en composants standards modulaires, et à générer automatiquement sur le plan du programme le logiciel de communication. Un système d'aide à la conception de logiciels de communication a été ainsi conçu et réalisé.

Le système est destiné à assister un opérateur non informaticien dans la conception et la réalisation de logiciels de communication fonctionnant sur un micro-ordinateur du marché. Il permet de diminuer l'effort et le temps de développement d'un tel logiciel, et par conséquent de réduire le coût du logiciel.

Les travaux décrits dans ce mémoire portent sur l'étude, la conception et la mise en oeuvre du système. L'accent est mis sur deux points :

- La définition et la structuration d'un système de communication général et adaptatif : elles rendent possible la génération automatique de logiciels de communication;
- La mise en oeuvre du système de génération, basé sur la structure générale du système de communication, et suivant l'approche "logiciels individualisables".

Le premier chapitre de ce mémoire résume les généralités dans l'aide à la communication pour handicapés : les besoins de systèmes de communication, le principe de la communication assistée, les composants matériels d'un système de communication, les fonctions de base de la communication, et la définition des constituants primaires. L'idée est d'en dégager un système de communication général et facilement individualisable.

Le deuxième chapitre présente la structuration et la conception de la partie logicielle du système de communication général. Il s'agit de la définition et de la structuration des données de base utilisées dans la communication, de la généralisation de la procédure de communication et de la conception des fonctions de base du logiciel. Cela permet de définir un noyau logiciel adaptatif pour le système de communication.

Au troisième chapitre, dans un premier temps, un rappel des approches développées dans le domaine de la production automatique de logiciels est effectué et l'une de ces approches : "logiciels individualisables" est analysée en détail. Ensuite, l'étude d'un système d'aide à la conception de logiciels de communication est faite. Il s'agit d'un logiciel individualisable d'aide à la communication. L'architecture et le fonctionnement sont décrits.

Nous présentons enfin dans le dernier chapitre les implémentations effectuées au cours de la poursuite du projet pour différentes configurations. La démarche de la génération d'un logiciel et la communication à l'aide du logiciel généré sont illustrées par quelques exemples d'utilisation du système.

CHAPITRE I :

**GENERALITES SUR L'AIDE A LA COMMUNICATION
POUR HANDICAPES**

Introduction

Les handicapés moteurs ou moteurs cérébraux sont souvent incapables de communiquer normalement avec leur monde extérieur. Ils ont besoins de systèmes d'aide à la communication, ou communicateurs.

Un système d'aide à la communication est un dispositif effectuant des acquisitions et transcodages d'informations, permettant à un opérateur de former des messages d'une façon qui lui est adaptée et dans un langage qui lui est propre, et de passer à un destinataire (humain ou machine) ces messages sous une forme compréhensible [TOU. 83]. Il doit donc être doté d'un ou plusieurs organes d'accès (interfaces d'entrée) adaptés, de moyens matériels et logiciels de mémorisation et de traitement d'informations, et d'interfaces de sortie connectables avec l'environnement de la communication [MYE. 82].

Dans cette première partie du mémoire, nous nous proposons d'abord de présenter les généralités sur l'aide à la communication pour handicapés :

- besoins du système d'aide à la communication,
- structure et fonctions du système d'aide à la communication,
- définitions des constituants (ou éléments) primaires de la communication,
- définition de l'environnement de la communication.

Ensuite, nous étudions de manière générale les différentes fonctions du système d'aide à la communication :

- sélection des constituants primaires,
- mémorisation des messages primaires,
- visualisation des informations,
- accélération à la construction des messages primaires,
- transformation des messages primaires en messages secondaires,
- émission des messages.

I.1 Besoins de systèmes d'aide à la communication pour handicapés

Pour des handicapés moteurs ou moteurs cérébraux, privés ou non de la parole, se pose toujours le problème de la communication avec leur monde extérieur. Privés de moyens efficaces de communication, les handicapés ont beaucoup de difficultés à se faire comprendre, à s'occuper eux-mêmes de leur vie quotidienne. Ils vivent en dépendant de leur entourage ou de la société, ce qui pèse beaucoup tant sur le plan économique que sur le plan humain ou moral [AYL. 83]. La communauté des handicapés s'agrandit malheureusement avec les accidents de la route. Il y a donc la nécessité, pour les aider, de concevoir et de réaliser des systèmes d'aide à la communication, appelés communicateurs.

Ceux-ci doivent permettre aux handicapés :

- de s'exprimer oralement ou par écrit,
- de résoudre les problèmes de la vie quotidienne [RAM. 79],
- et d'accéder, le mieux possible, à une activité scolaire ou professionnelle [NEL. 83], [GRO. 83].

En tenant compte de la spécificité de chaque handicapé, tout communicateur doit :

- s'adapter à son utilisateur, ce qui signifie une manipulation aisée et un apprentissage facile au point de vue physique et mental,
- être modifiable et extensible pour suivre l'évolution constante de l'environnement de la communication, et celle de la capacité de communication de son utilisateur,
- être configurable en fonction des ressources financières de son utilisateur, ce qui implique une construction modulaire,
- et avoir un prix accessible par une conception et une fabrication à partir de composants et de matériel standards commercialisés [BLA. 81].

I.2 Système d'aide à la communication pour handicapés

I.2.1 Comportement extérieur d'un système d'aide à la communication [JAF. 85]

Vu de l'extérieur, un système d'aide à la communication pour handicapés sert d'intermédiaire de communication entre l'utilisateur et l'environnement. Il a pour mission de "dialoguer" d'une part avec l'utilisateur (interaction handicapé-communicateur) et d'autre part avec l'environnement (interaction communicateur-environnement), comme le montre la figure 1.1.

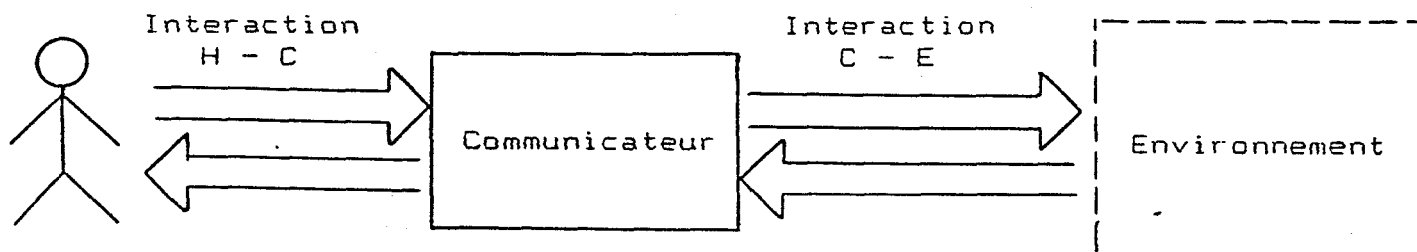


Fig. 1.1 Comportement extérieur du communicateur

L'utilisateur construit grâce au communicateur un message en sélectionnant, à l'aide d'une interface d'accès et d'une méthode de sélection, des éléments parmi un ensemble de base prédéfini. En même temps le communicateur visualise sur un dispositif d'affichage le message en cours de construction, et facilite à l'utilisateur l'accès aux éléments les plus probables. Le communicateur peut éventuellement transmettre à l'utilisateur des messages envoyés par l'environnement.

Le communicateur émet par ailleurs à l'adresse de l'environnement le message, émis par l'utilisateur, après d'éventuelles transformations, pour être compréhensible par le destinataire matériel ou humain.

I.2.2 Fonctions d'un système d'aide à la communication [TOU. 83]

Un système d'aide à la communication comprend d'une manière générale les fonctions suivantes :

- acquisition et décodage des interactions de l'utilisateur,
- mémorisation des messages,
- visualisation des messages, des éléments de base à sélectionner et diverses informations,
- accélération de la composition des messages à partir des éléments de base,
- transformation des messages de la forme primaire en une forme compréhensible par l'environnement,
- émission des messages transformés vers l'environnement.

Un schéma fonctionnel du communicateur est proposé à la figure 1.2.

Ces fonctions vont être analysées en détail par la suite.

I.2.3 Eléments matériels d'un système d'aide à la communication [ROS. 82] [TOU. 83]

Pour pouvoir effectuer les fonctions énumérées, un système d'aide à la communication doit être doté des constituants suivants :

- un ou plusieurs organes d'accès convenablement adaptés à l'utilisateur handicapé, et parfaitement interfacés,
- un processeur pouvant effectuer des transcodages et des manipulations d'informations,
- des dispositifs de visualisation pour affichage des éléments à sélectionner et des éléments sélectionnés,
- des moyens pour mémoriser les messages ou des dictionnaires,
- et des interfaces et des organes de sortie adaptés à l'environnement.

La figure 1.3 montre ces composants matériels et les interactions entre eux, l'utilisateur et l'environnement.

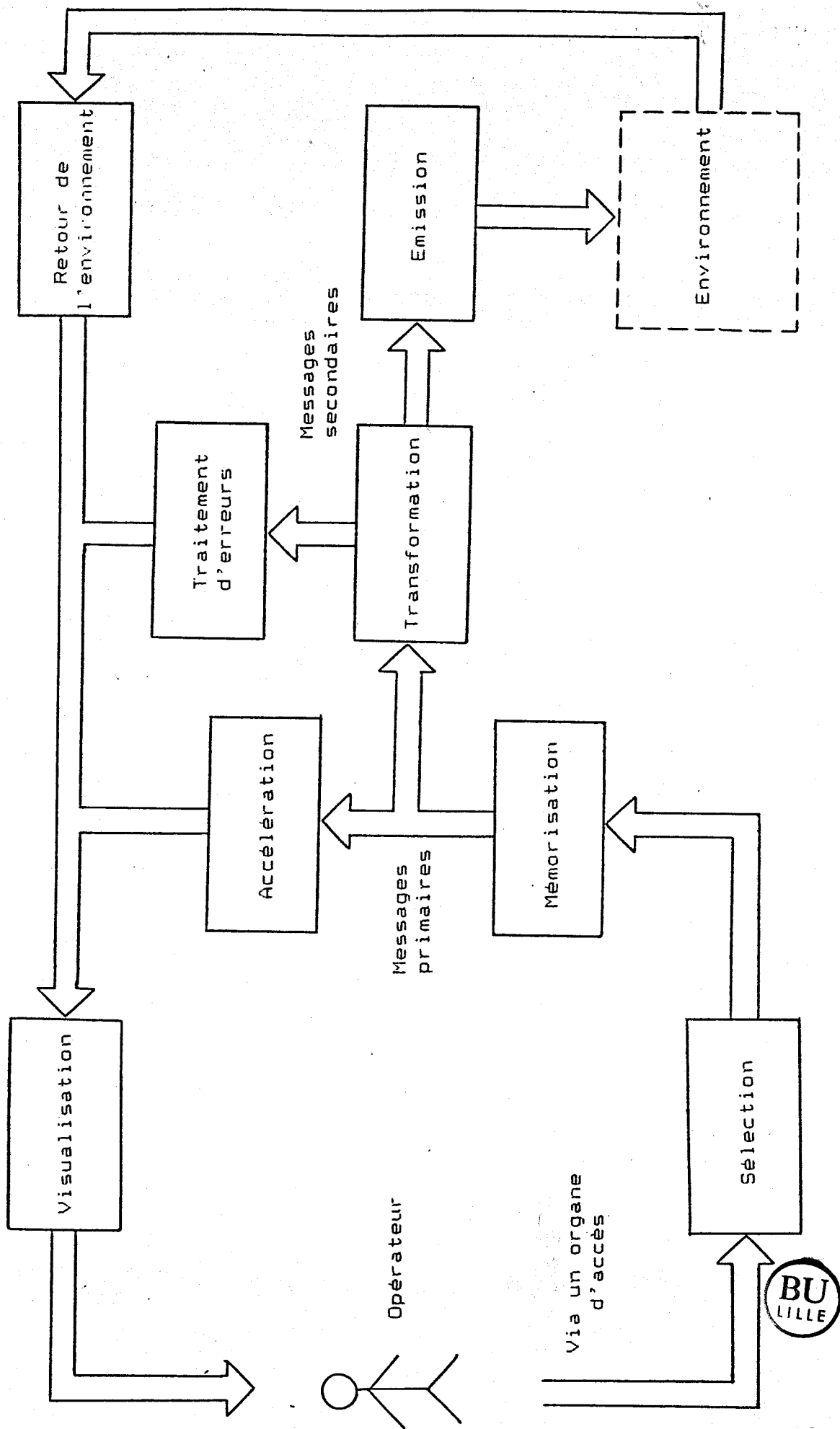


Fig. 1.2 Fonctionnement du communicateur

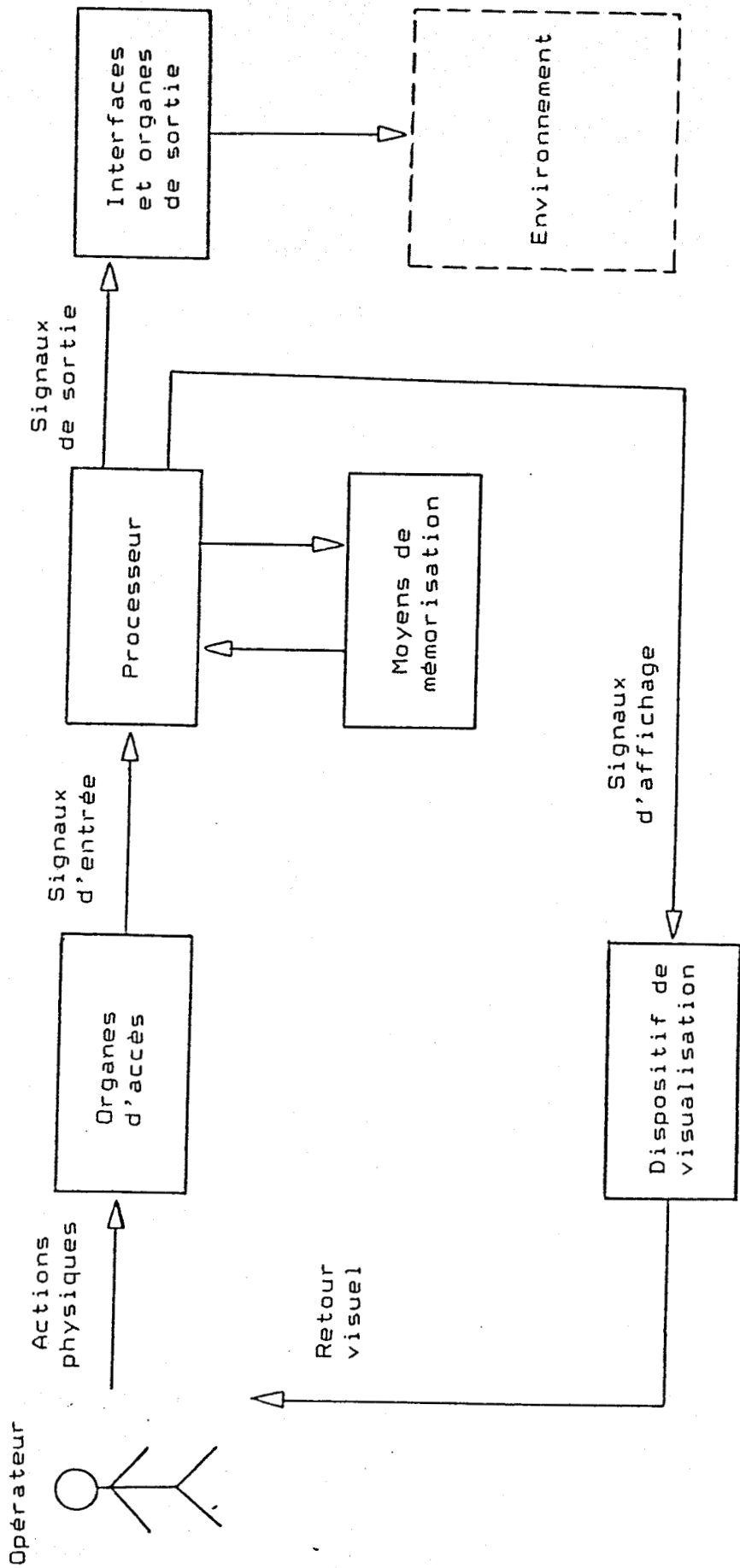


Fig. 1.3 Composants du communicateur et leurs interactions



I.3 Environnement de la communication

L'environnement d'une personne handicapée comprend :

- des êtres humains : parents, amis, enseignants, collègues, ...
- des machines générales à usage domestique ou professionnel : Minitel, terminal d'ordinateur [NEL. 83], synthétiseur de voix, ...
- des dispositifs spéciaux pour handicapés : robot-manipulateur [SCH. 81] [GUI.76], fauteuil roulant, baignoire, ...

I.4 Eléments (ou constituants) primaires dans un communicateur [TOU. 83]

Dans un communicateur, il existe un ensemble d'éléments prédéfinis et sélectionnables par l'opérateur. Ces éléments sont appelés constituants primaires ou éléments de base.

Au cours de la communication, ces éléments sont présentés à l'opérateur sur un dispositif d'affichage pour être sélectionnés à l'aide d'un organe d'accès et d'une méthode de sélection. La visualisation d'un élément est appelé pavé, car elle est généralement carré ou rectangulaire.

Ces éléments sont utilisés soit pour composer des messages, soit pour commander une action. On distingue donc deux catégories d'éléments : les éléments informatifs (appelés aussi pavés affichables) et les éléments de commande (appelés aussi pavés exécutable).

Ces éléments sont logiquement classés en groupes (menus) et sous-groupes suivant différents critères.

I.4.1 Eléments informatifs (pavés affichables)

Les éléments informatifs sont appelés pavés affichables du fait que tout élément informatif sélectionné correspond à un affichage dans un message.

Dans les communicateurs, on rencontre des éléments informatifs très divers. On peut citer de façon non exhaustive les catégories suivantes :

- les lettres d'une langue naturelle,
- les chiffres,
- des caractères spéciaux (signes de ponctuation, opérateurs mathématiques, et divers symboles),
- des mots,
- des phrases,
- des phonèmes
- les mots réservés d'un langage informatique de programmation [NEL. 83],
- des primitives pictographiques (point, carré, cercle, couleur, etc),
- les composants des symboles BLISS,

- les éléments du PAR-LE-SI-LA-B,
- les symboles de notation musicale, ...

Un élément informatif peut éventuellement avoir un comportement particulier. On trouve ainsi :

- Le pavé modulateur

Un pavé modulateur est un pavé affichable dont l'existence dans un message détermine celle de certains pavés qui le suivent, de sorte que la suppression du pavé entraîne celle de ces derniers. Par exemple, si un message est constitué des trois pavés : vouloir boire jus d'orange, il est logique que la suppression du pavé boire du message entraîne la suppression du pavé jus d'orange;

- Le pavé générateur

Un pavé générateur est un pavé affichable dont la sélection entraîne celle d'autres pavés affichables. C'est à dire qu'il génère ces pavés. A titre d'exemple, le pavé boire peut engendrer le message je voudrais boire quelque chose;

- Le pavé modulateur-générateur

Un pavé modulateur-générateur est un pavé affichable qui génère d'autres pavés, et qui module en même temps ceux-ci. Considérons l'exemple précédent : la suppression du pavé boire du message engendré peut conduire à la suppression du message;

- Le pavé affichable-exécutable

Un pavé affichable-exécutable est un pavé affichable dont la sélection déclenche une commande supplémentaire. Par exemple, la sélection du pavé boire peut entraîner, en plus de la mémorisation du message je voudrais boire quelques choses dans le tampon courant, l'émission immédiate du message vers un synthétiseur vocal;

- Le pavé vide

Un pavé vide représente un espace blanc de dimension fixe ou initialement non déterminée. Il sert comme séparateur dans la représentation des messages primaires.

I.4.2 Eléments de commande (pavés exécutables)

Les éléments de commande se distinguent des éléments informatifs par le fait qu'il n'ont pas pour mission d'être des composants de messages. La sélection d'un tel élément déclenche immédiatement l'exécution de la commande correspondante.

On peut classer les éléments de commande d'un communicateur en deux groupes :

- Les éléments de contrôle d'édition des messages :

Ces sont les commandes utilisées pour faciliter le processus d'édition des messages primaires, ou bien pour manipuler le message en élaboration. Elles sont analogues aux commandes d'un éditeur de texte. Les commandes peuvent exister pour :

- le déplacement du curseur dans le tampon de message en cours,
- l'insertion, la suppression et l'interversion de pavés sélectionnés,
- le changement de tampon de message,
- le changement de fenêtre de visualisation de message,
- le changement de menu,
- l'ajustement de la vitesse de balayage,
- le changement de mode de travail.

Ces commandes n'ont pas d'interactions avec l'environnement de la communication.

- Les commandes de contrôle de l'environnement

Il s'agit ici des commandes qui contrôlent les dispositifs de l'environnement par l'intermédiaire des interfaces de sortie. Les commandes sont très diverses. On trouve par exemple : allumage d'un téléviseur, sortie du message d'un tampon vers un synthétiseur vocal ou une imprimante, ...

Souvent, ces commandes impliquent l'usage de transformation du message primaire en message final en fonction du dispositif destinataire.

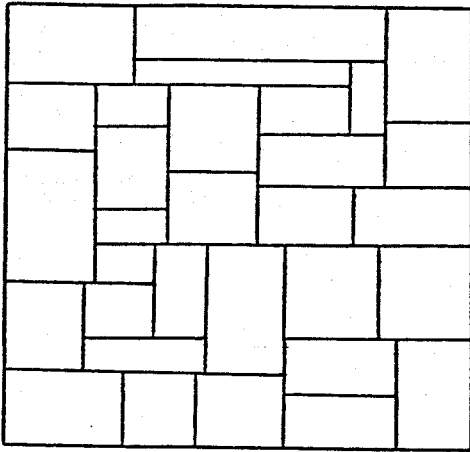
I.4.3 Organisation logique des constituants primaires

Les constituants primaires sont logiquement organisés en menus et sous-menus . Un ensemble de constituants (pavés) constitue un menu (ou sous-menu), c'est à dire une page affichable, qui comporte par ailleurs :

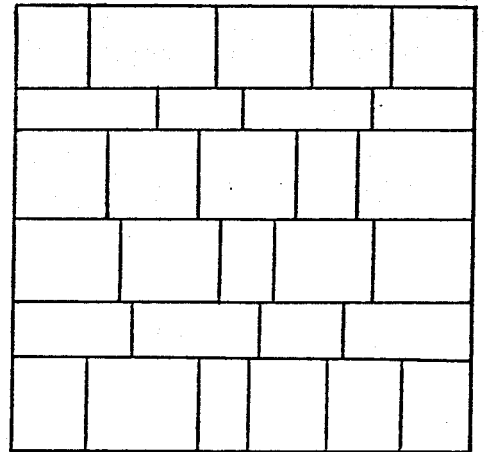
- la façon dont l'utilisateur peut sélectionner l'un de ces constituants;
- le mode alphanumérique ou graphique d'affichage;
- la façon dont le communicateur désigne les constituants sélectionnables, et qui peut être l'un des moyens suivants :
 - inverse-vidéo,
 - clignotement,
 - soulignage,
 - déplacement du curseur,
 - encadrement,
 - changement de couleur;
- la disposition des constituants dans une fenêtre d'affichage : ceci peut se faire suivant l'un des quatre cas de la figure 1.4.

La classification des constituants en menus et sous-menus se fait suivant divers critères comme par exemple :

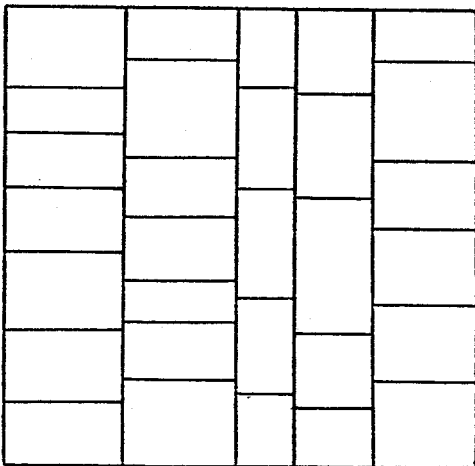
- suivant des catégories syntaxiques comme par exemple : PRONOM, VERBE,...
- suivant des catégories sémantiques, par exemple BOISSON, ANIMAL, ACTION, ...



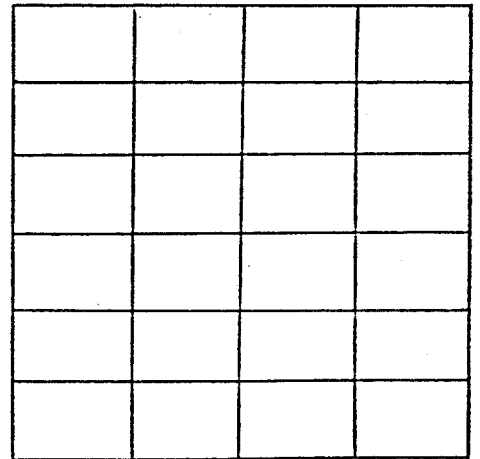
(1) Structure "mosaïque"



(2) Structure "ligne"



(3) Structure "colonne"



(4) Structure "matrice"



Fig. 1.4 Structuration des pavés dans un menu

- selon la fonction des éléments, par exemple, lettres, commandes d'édition, commandes de contrôle de l'environnement, ...

L'utilisation des menus doit se faire de façon à accélérer la communication, en tenant compte des habitudes et des capacités physiques et mentales du handicapé, dans la limite des possibilités d'affichage.

I.5 Sélection des constituants primaires

Pour communiquer, l'utilisateur sélectionne des constituants primaires pour construire un message primaire ou pour exécuter des commandes. Ceci se fait grâce à une interface (ou organe) d'accès et à une méthode de sélection.

I.5.1 Interfaces (ou organes) d'accès adaptées

Les organes d'accès doivent être parfaitement adaptés à chaque type de handicap. Aussi est-il nécessaire d'évaluer dans chacun des cas toutes les grandeurs physiques contrôlables par le handicapé et de trouver le moyen de les transformer en signal informatif par des interfaces appropriées.

On cherche ainsi à utiliser tout mouvement, tout signal musculaire, le souffle, le mouvement des yeux, la modulation glottique, ... [MYE. 82] [ROS. 82]

Malgré la diversité des organes d'accès, on peut distinguer les trois classes suivantes :

- interrupteur tout-ou-rien unique,
- multi-interrupteur,
- commande positionnelle.

I.5.2 Méthodes de sélection des constituants [TOU. 83] [ROS. 82] [HAN. 85] [THO. 81]

La désignation d'un pavé dans un menu se fait à l'aide d'une méthode de sélection qui consiste à décoder les informations venues de l'interface d'accès et à repérer le pavé, ainsi à le sélectionner.

Il existe plusieurs méthodes de sélection applicables qu'il est possible de classer suivant l'une des catégories suivantes :

- sélection directe sans codage (sélection positionnelle),
- sélection avec codage,
- sélection par balayage.

Il n'y a pas de méthode de sélection idéale. Il faut choisir pour chaque individu handicapé la mieux adaptée aux possibilités physiques ou mentales.

I.5.2.1 Sélection directe sans codage

Il s'agit d'une méthode de sélection positionnelle qui ne demande aucun codage intermédiaire. L'utilisateur sélectionne un pavé en pointant, avec une tige buccale par exemple, la touche ou la position correspondante sur l'interface d'accès, qui peut être un multi-interrupteur, une table à digitaliser, ou une matrice de capteurs opto-électroniques.

Cette méthode permet la sélection d'un pavé à chaque action de l'interface d'accès, ce qui signifie un temps de sélection très court. Par contre, elle exige un effort physique important, nécessaire pour le pointage sur un clavier, ou un système analogue, ayant au moins autant de touches que le nombre de pavés sélectionnables.

I.5.2.2 Sélection avec codage

A l'aide de la méthode de sélection avec codage intermédiaire, l'utilisateur choisit un pavé en composant le code correspondant. Par exemple, avec le codage ligne-colonne, le code est composé du numéro de la ligne où se trouve le pavé à sélectionner, et du numéro du pavé à l'intérieur de la ligne.

Cette méthode permet l'utilisation d'un clavier dont le nombre de touches est moins important que la sélection directe sans codage, ce qui conduit à une diminution de l'effort physique nécessaire à la manipulation, mais à une augmentation du temps de communication, par rapport à la sélection directe sans codage. Plus le niveau de codage est profond, moins l'effort physique est demandé, mais plus le temps de communication est long.

Les codages les plus utilisés dans l'aide à la communication sont le codage ligne-colonne (ou colonne-ligne), le codage dichotomique, le codage ETAN, le code MORSE, ...

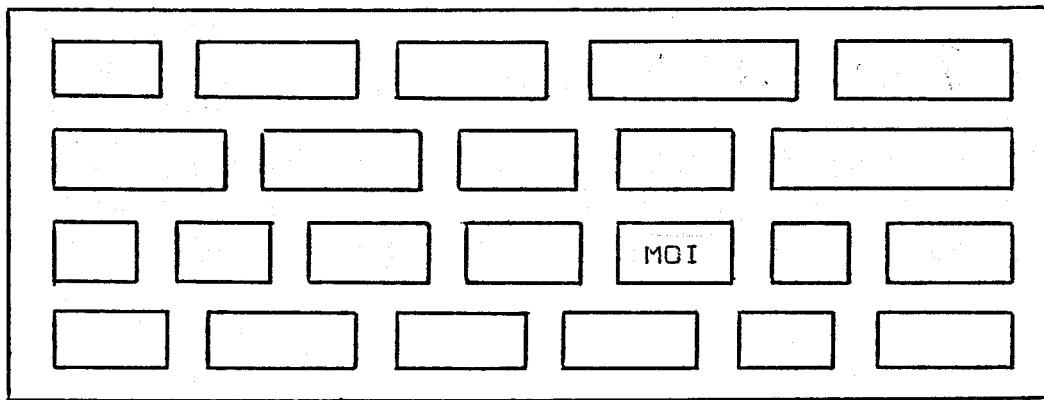
1) Sélection avec codage ligne-colonne (ou colonne-ligne)

Avec le codage ligne-colonne (ou colonne-ligne), on effectue un codage à deux niveaux. On groupe les pavés d'un menu en lignes (ou colonnes), et on numérote les lignes (ou colonnes) ainsi que les pavés à l'intérieur de chaque ligne (ou colonne).

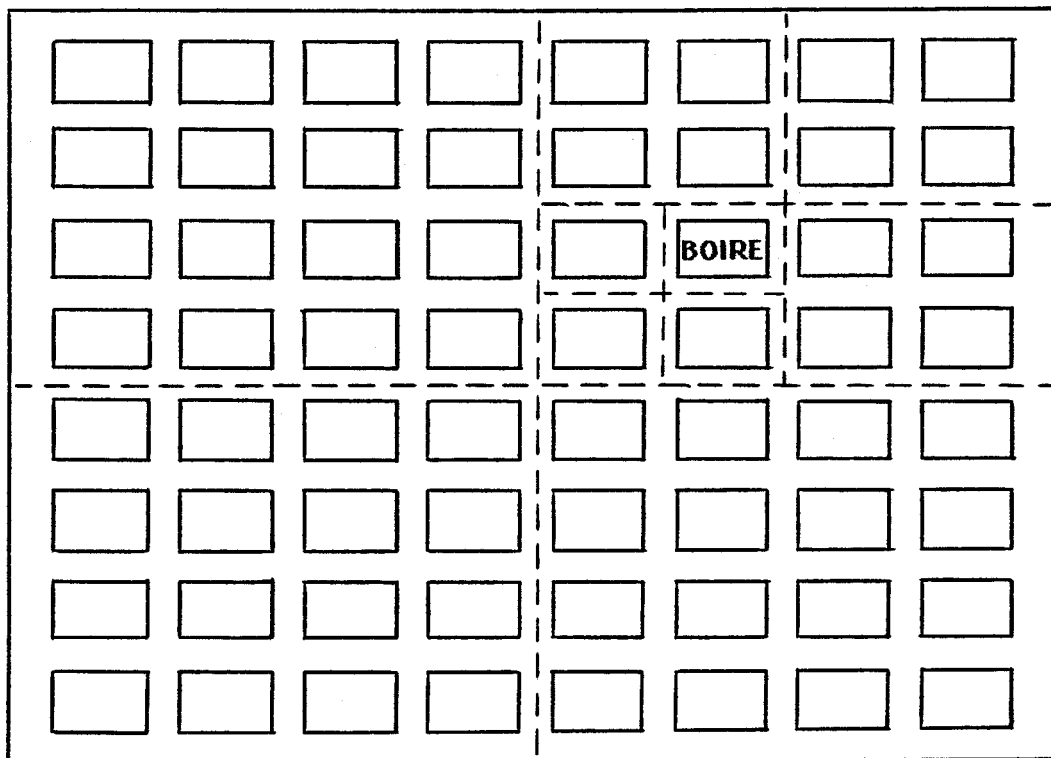
La sélection d'un pavé se fait par la spécification du numéro de la ligne (ou colonne) où le pavé figure, et le numéro du pavé à l'intérieur de celle-ci.

Par exemple, dans la figure 1.5(1), pour sélectionner le pavé MOI qui est le cinquième pavé de la troisième ligne, il faut composer les codes 3 et 5.

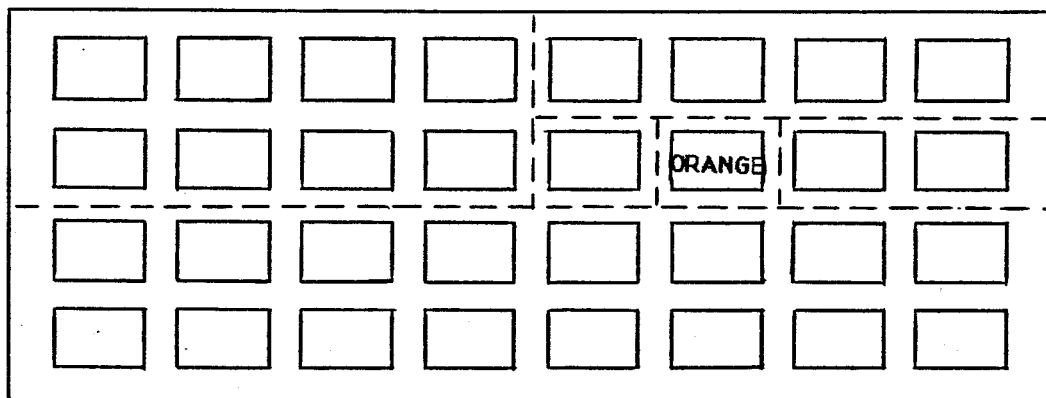
Ce mode de codage est le plus utilisé, car il est simple et naturel. Il demande néanmoins à l'utilisateur de manipuler un clavier avec un nombre assez important de touches.



(1) Forme ligne-colonne



(2) Forme matricielle 8 x 8



(3) Forme matricielle 4 x 8



Fig. 1.5 Exemples de sélection de pavés

2) Sélection avec codage dichotomique

Dans le cas où les pavés à sélectionner sont au nombre de la puissance de deux d'un entier pair, soit $n = 2^{2m}$ où n le nombre des pavés, m un entier positif, on peut alors appliquer le codage dichotomique.

Il s'agit d'un codage à m niveaux. L'ensemble des pavés est divisé en quatre groupes avec un nombre égal de pavés, et chacun de ces groupes est lui-aussi divisé de la même façon, et ainsi de suite, ... Au niveau le plus bas de la division, chaque sous-groupe contient un seul pavé. La figure 1.5(2) présente une matrice carrée de 64 pavés, qui illustre les $3 (= 0,5 \log_2 64)$ niveaux de division.

Pour sélectionner un pavé parmi $n (= 2^{2m})$ pavés, il faut suivre les m étapes de sélection suivantes :

- 1^{ière} étape :
La sélection de l'un des quatre groupes ayant chacun $2^{2(m-1)}$ pavés, dans lequel figure le pavé désiré;
- 2^{ème} étape :
La sélection de l'un des quatre sous-groupes, ayant chacun $2^{2(m-2)}$ pavés, du groupe précédemment sélectionné;
- ...
- m ^{ème} étape :
La sélection du pavé désiré parmi les quatre pavés du sous groupe précédemment sélectionné.

La sélection d'un groupe parmi quatre peut se faire en codage ligne-colonne ou par positionnement.

Par exemple, la sélection du pavé BOIRE de la figure 1.5(2) se fait en introduisant successivement les codes 1 2, 2 1 et 1 2 si le codage de groupes s'effectue en ligne-colonne, ou les codes 2, 3, et 2 si la sélection positionnelle est utilisée.

Il est préférable de permettre à l'utilisateur de pouvoir annuler à toute étape intermédiaire la sélection précédente, et revenir à l'étape plus haute, afin de corriger les erreurs de sélection. Pour ce faire, une touche ou un code de validation est alors nécessaire.

Le codage dichotomique permet l'utilisation d'un clavier avec deux ou quatre touches comme organe d'accès, par conséquent un effort faible de manipulation. Il nécessite néanmoins une suite de codes plus importante que le codage ligne-colonne, ce qui présente un temps de communication plus long. En plus, il exige que le nombre des pavés présentés soit 2^{2m} . Cette condition ne peut pas être toujours satisfaite dans la pratique. Le codage dichotomique mixte permet de palier ce problème.

3) Sélection avec codage dichotomique mixte

Le codage dichotomique mixte demande que le nombre des pavés à sélectionner soit une puissance de deux, c'est à dire $n = 2^m$ où n est le nombre des pavés, m un entier positif.

Il s'agit d'un codage à m niveaux. Différent par rapport au vrai codage dichotomique, le codage dichotomique mixte divise l'ensemble des pavés à sélectionner en deux groupes, au lieu de quatre, avec un nombre égal de pavés, et chaque groupe en deux sous-groupes, etc, ... Au niveau le plus bas de la division, chaque sous-groupe n'a qu'un seul pavé. La figure I.5(3) présente 32 pavés casés dans une matrice de 4×8 .

Analogue à la sélection avec codage dichotomique, la sélection avec codage mixte nécessite, pour sélectionner un pavé parmi $n (= 2^m)$ pavés, m étapes de sélection. Dans chaque étape, on sélectionne l'un des deux sous-groupes du groupe en cours en introduisant le code correspondant.

Pour sélectionner le pavé ORANGE de la figure 1.5(3), par exemple, il faut composer successivement les codes 1, 2, 2, 1 et 2.

Le codage dichotomique mixte permet l'utilisation d'un clavier avec deux touches de sélection, et la condition exigée par ce codage est plus facile à satisfaire que celle du vrai codage dichotomique.

Le codage dichotomique mixte applique en même temps le vrai codage dichotomique et le codage ligne-colonne, d'où vient son nom.

I.5.2.3 Sélection par balayage

Pour les handicapés ayant une capacité motrice très réduite, la méthode de sélection par balayage permet d'effectuer la sélection de constituants primaires à l'aide d'un simple (ou double) interrupteur adapté.

L'ensemble des pavés présentés est initialement divisé en groupes, sous-groupes, sous-sous-groupes, ... Commenant par le niveau le plus haut de la division, le balayage se fait groupe par groupe à l'aide d'un moyen visuel de désignation, par l'inverse-vidéo par exemple. Un appui sur l'organe d'accès conduit à la sélection du groupe désigné au moment de l'appui. Si le groupe contient des sous-groupes, on reprend le balayage à l'intérieur du groupe. Et ainsi de suite, jusqu'à la sélection d'un pavé.

Si l'on utilise un simple interrupteur comme organe d'accès, le balayage s'effectue automatiquement. Sa vitesse (la durée de temps d'un pas) peut être ajustée par l'utilisateur au moyen d'un élément de commande.

Dans le cas d'utilisation d'un double interrupteur, une touche sert à la sélection, l'autre à la validation du balayage de façon répétitive ou continue.

Suivant la façon de grouper les pavés à sélectionner, on distingue les modes suivants :

- balayage en série pavé par pavé,
- balayage ligne-colonne (ou colonne-ligne),
- balayage dichotomique et
- balayage dichotomique mixte.

1) Sélection par balayage pavé par pavé

Dans ce cas, on a un seul niveau de groupement de l'ensemble des pavés à sélectionner. Le balayage se fait pavé par pavé de façon séquentielle, et suivant l'ordre naturel des pavés, i.e. ligne par ligne (ou colonne par colonne).

Une seule frappe de la touche de sélection suffit pour la sélection d'un pavé. Le temps d'attente nécessaire à la sélection dépend de la position du pavé dans l'ensemble. La sélection du $i^{\text{ème}}$ pavé dans l'ensemble demande au moins $(i-1)$ pas de balayage.

Supposons qu'il y ait n pavés à balayer, dans l'ordre : $P_1, P_2, P_3, \dots, P_n$, et que leurs probabilités d'utilisation soient respectivement $p_1, p_2, p_3, \dots, p_n$, avec

$$\sum_{i=1}^n p_i = 1 \text{ et } p_i > 0,$$

on peut calculer le temps moyen d'attente pour une sélection :

$$\sum_{i=1}^n [(i-1) * p_i] t \quad \text{avec } t : \text{ le temps d'un pas de balayage,}$$

- dans le cas où $p_i = 1/n$ avec $i = 1, 2, \dots, n$, c'est à dire à égalité des probabilités d'utilisation, on a le temps moyen

$$\sum_{i=1}^n (i-1)/n = [0,5 * (N-1)] t \quad \text{avec } t : \text{ le temps d'un pas de balayage,}$$

il est proportionnel à n , le nombre des pavés à sélectionner;

- dans le cas où $p_i > p_{i+1}$ ($i=1, 2, \dots, n$), on peut démontrer que

$$\sum_{i=1}^n (i-1) * p_i < 0,5 * (N-1).$$

Ce dernier cas nous conduit à balayer les pavés suivant l'ordre décroissant des fréquences d'utilisation, dans le but de diminuer le temps moyen d'attente, et donc d'accélérer la communication.

2) Sélection par balayage ligne-colonne (ou colonne-ligne) [TIC. 85]

En sélection par balayage ligne-colonne (ou colonne-ligne), les pavés à balayer sont groupés en lignes (ou colonnes) de pavés. C'est un groupement à deux niveaux. Dans un premier temps, le balayage commence ligne par ligne (ou colonne par colonne), jusqu'à la sélection d'une ligne (ou colonne) à la suite d'un appui sur l'organe d'accès. Il se poursuit alors par un balayage pavé par pavé à l'intérieur de la ligne (ou colonne) sélectionnée. Ce balayage aboutit à la sélection d'un pavé.

Il faut deux appuis de la touche de sélection pour une sélection de pavé, donc un peu plus d'effort physique par rapport au balayage pavé par pavé. Le temps d'attente est néanmoins considérablement réduit. La sélection du $j^{\text{ème}}$ pavé de la $i^{\text{ème}}$ ligne (ou colonne) demande $\{(i-1)+(j-1)\}$ pas de balayage.

Considérons le cas spécial où les pavés à balayer sont représentés par une matrice d'une dimension $m \times n$, c'est à dire m lignes (ou colonnes) de n pavés. La figure 1.6 présente une telle matrice avec, pour chaque case, le nombre de pas nécessaire pour accéder à la case en balayage ligne-colonne.

0	1	2	n-2	n-1
1	2	3	n-1	n
2	3	4	n	n+1
...
...	$i+j-2$
...
m-2	m-1	m	m+n-4	m+n-3
m-1	m	m+1	m+n-3	m+n-2

Fig. 1.6 $m \times n$ matrice avec les temps en balayage ligne-colonne



Le temps moyen d'accès à un pavé s'exprime par la formule

$$\sum_{i=1}^m \sum_{j=1}^n p_{ij} * [(i-1)+(j-1)] \quad t$$

où t est le temps d'un pas de balayage et p_{ij} la probabilité d'utilisation du $j^{\text{ème}}$ pavé de la $i^{\text{ème}}$ ligne de la matrice $m \times n$ avec

$$p_{ij} > 0 \text{ et } \sum_{i=1}^m \sum_{j=1}^n p_{ij} = 1$$

avec $i = 1, 2, \dots, m$ et $j = 1, 2, \dots, n$.

Si tous les $m \times n$ pavés ont la même chance d'être choisis, soit $p_{ij} = 1/(m \times n)$ avec $i = 1, 2, \dots, m$ et $j = 1, 2, \dots, n$, la formule devient :

$$\sum_{i=1}^m \sum_{j=1}^n [(i-1)+(j-1)] / (m \times n) = [0,5 \times (m+n) - 1] t$$

avec t : le temps d'un pas de balayage.

En examinant le résultat ci-dessus et les temps d'accès aux cases, indiqués dans la figure 1.6, nous pouvons conclure que pour réduire au minimum le temps de sélection, il y a intérêt :

- à ranger les pavés à balayer dans une matrice la plus carrée possible, ou à utiliser la demi-matrice supérieure;
- et à disposer les pavés dans la matrice de manière à ce que plus un pavé est utilisé, plus il se trouve proche du coin haut-gauche de la matrice.

3) Sélection par balayage dichotomique

La sélection par balayage dichotomique consiste à grouper les 2^{2m} pavés à sélectionner en m niveaux comme pour le codage dichotomique, et à sélectionner dans chaque niveau l'un des quatre sous-groupes du groupe en cours par un balayage, jusqu'à la sélection du pavé désiré. La sélection d'un pavé parmi $n (= 2^{2m})$ nécessite alors m niveaux de balayage.

Le balayage dans un niveau peut s'effectuer de manière séquentielle groupe par groupe ou ligne-colonne.

Dans le premier cas, les quatre sous-groupes du groupe en cours sont balayés sous-groupe par sous-groupe. Un appui de la touche de sélection permet de sélectionner un sous-groupe. La figure 1.7(1) présente 64 pavés casés dans une matrice carrée 8×8 , avec les temps d'accès à chaque case. On peut démontrer que le temps moyen d'accès à un pavé est $0,75 \log_2 n (= 1,5 \times m)$, sous la condition d'égalité des fréquences d'utilisation des n pavés.

Dans le second cas, les quatre sous-groupes du groupe en cours sont balayés de façon ligne-colonne. Il faut donc deux appuis de la touche de sélection pour sélectionner un sous-groupe. La figure 1.7(2) présente 64 pavés casés dans une matrice carrée 8×8 , avec les temps d'accès à chaque case. Il est démontrable que sous la condition d'égalité des fréquences d'utilisation des pavés et dans le cas où les $n (= 2^{2m})$ pavés sont rangés dans une matrice carrée de dimension $2m \times 2m$, le temps moyen d'accès à un pavé est $0,5 \log_2 n (= m)$ unités.

De manière générale, la sélection par balayage dichotomique exige de l'utilisateur une capacité physique plus importante que le balayage en série et le balayage ligne-colonne (ou colonne-ligne), du fait qu'il faut m (ou $2m$) frappes de la touche de sélection pour la sélection d'un pavé parmi $n (= 2^{2m})$ pavés. Elle permet par contre un gain considérable en temps d'accès. Dans le cas de non égalité des fréquences d'utilisation des pavés, un rangement étudié des pavés peut réduire le temps moyen d'accès.

Comme la sélection avec codage dichotomique, le balayage dichotomique nécessite que les pavés à sélectionner soient au nombre de 2^{2m} . Si cette exigence ne peut pas être satisfaite, on peut faire appel au balayage dichotomique mixte.

4) Sélection par balayage dichotomique mixte

La sélection par balayage dichotomique mixte demande que le nombre des pavés sélectionnables soit une puissance de deux, soit $n = 2^m$ avec n le nombre des pavés et m un entier positif.

L'ensemble des n pavés sélectionnables est dichotomisé, comme pour la sélection avec codage dichotomique mixte, en groupes et sous-groupes sur m niveaux. La sélection se fait de manière à sélectionner chaque fois une moitié de pavés du groupe en cours. La sélection d'un pavé nécessite m étapes progressives de balayage. Chaque étape de balayage consiste à balayer les deux sous-groupes du groupe en cours. Une frappe de la touche de sélection permet de sélectionner l'un des deux sous-groupes.

Comme la sélection par balayage dichotomique, la sélection par balayage dichotomique mixte augmente la vitesse de communication du fait que le temps moyen d'accès est $0,5 \log_2 n$ unités sous la condition d'égalité des fréquences d'utilisation des pavés, et demande par contre à l'opérateur une capacité physique plus importante que le balayage en série et le balayage ligne-colonne (ou colonne-ligne) : il faut m frappes de la touche de sélection pour la sélection d'un pavé parmi $n (= 2^m)$ pavés. Dans le cas de non égalité des fréquences d'utilisation, un rangement étudié des pavés suivant les fréquences peut réduire le temps moyen d'accès, ce qui peut être justifié par l'exemple de la figure 1.7(3) qui montre une matrice 4×8 avec les temps d'accès pour chaque case dans ce mode de balayage. Par ailleurs, la condition d'emploi de ce mode est plus facile à satisfaire que le mode de balayage dichotomique.

I.6 Mémorisation des messages primaires

Au cours de la communication, les messages primaires constitués des éléments informatifs sélectionnés sont temporairement mémorisés dans des tampons ou buffers de message.

Dans un communicateur, il peut exister autant de tampons de message que nécessaire. La dimension des tampons varie d'un système à l'autre. Cela dépend des besoins de l'utilisateur et de la quantité de mémoire disponible.

0	1	1	2	1	2	2	3
2	3	3	4	3	4	4	5
2	3	4	5	3	4	5	6
4	5	6	7	5	6	7	8
2	3	3	4	3	4	4	5
4	5	5	6	5	6	6	7
4	5	6	7	5	6	7	8
6	7	8	9	7	8	9	10

Fig. 1.7(1) 8 x 8 matrice avec les temps d'accès en balayage dichotomique et série

0	1	1	2	1	2	2	3
1	2	2	3	2	3	3	4
1	2	2	3	2	3	3	4
2	3	3	4	3	4	4	5
1	2	2	3	2	3	3	4
2	3	3	4	3	4	4	5
2	3	3	4	3	4	4	5
3	4	4	5	4	5	5	6

Fig. 1.7(2) 8 x 8 matrice avec les temps d'accès en balayage dichotomique et ligne-colonne

0	1	1	2	1	2	2	3
1	2	2	3	2	3	3	4
1	2	2	3	2	3	3	4
2	3	3	4	3	4	4	5

Fig. 1.7(3) 4 x 8 matrice avec les temps d'accès en balayage dichotomique mixte

Les tampons doivent pouvoir être visualisés sur un dispositif d'affichage, un écran vidéo par exemple, pour que l'utilisateur puisse voir ses messages. Plusieurs tampons peuvent éventuellement être visualisés en même temps.

Quand cela est possible, il est très utile de pouvoir sauvegarder les tampons de message sur un support extérieur de mémorisation, une disquette par exemple.

Pour l'utilisateur, un tampon de message est comme un papier sur lequel il peut écrire des messages, les modifier, les effacer, les copier et les conserver. A l'aide des éléments de commande, l'utilisateur peut contrôler constamment l'élaboration de ses messages, et effectuer les opérations suivantes :

- changement du tampon en cours,
- affichage du tampon,
- initialisation du tampon,
- sauvegarde du tampon,
- appel d'un tampon sauvegardé,
- chaînage de deux tampons, ...

I.7 Visualisation des informations dans l'aide à la communication

I.7.1 Visualisation des informations

Dans l'aide à la communication, l'interaction dans le sens communicateur-handicapé se réalise par visualisation sur un dispositif d'affichage :

- des constituants primaires,
- des messages primaires et
- de divers renseignements.

I.7.1.1 Visualisation des constituants primaires

Le communicateur doit pouvoir visualiser les constituants primaires afin que l'utilisateur puisse les sélectionner. La visualisation des constituants primaires se fait par menu. Un menu est en effet défini comme une page affichable de pavés. Il se visualise dans une zone rectangulaire sur un dispositif d'affichage.

A un instant donné, on a besoin de ne visualiser qu'un seul menu, défini comme le menu en cours, et on a accès seulement à ses éléments.

I.7.1.2 Visualisation des messages primaires

Le communicateur doit pouvoir visualiser sur un dispositif d'affichage les messages construits. L'utilisateur voit ses messages, et les manipule de façon interactive.

Les messages primaires étant stockés dans les tampons de message, la visualisation de ces messages se fait tampon par tampon. On visualise tout ou partie d'un tampon dans une fenêtre d'affichage qui lui est réservée. Plusieurs fenêtres de message peuvent exister simultanément si nécessaire et le dispositif d'affichage le permet.

La visualisation des tampons de message est contrôlée par l'utilisateur au moyen des éléments de commande de contrôle de la visualisation.

I.7.1.3 Visualisation de renseignements divers

En dehors des éléments primaires et des messages primaires, il est souvent nécessaire, dans le but d'informer l'utilisateur, de visualiser divers renseignements utiles, comme :

- la signalisation des erreurs,
- le numéro des tampons affichés,
- le taux d'occupation du tampon en cours,
- des informations d'aide, ...

Pour ces renseignements, on utilise une petite fenêtre spéciale d'affichage, dite fenêtre de renseignement.

I.7.2 Ecrans de visualisation des informations

Dans l'aide à la communication pour handicapés voyants, on utilise, dans la plupart des cas, un seul écran à tube cathodique pour visualiser les types d'informations mentionnées ci-dessus. L'écran est donc partagé en fenêtres : une fenêtre de menu, une ou plusieurs fenêtres de message, et une fenêtre de renseignement.

Les dispositions des fenêtres sur un seul écran ainsi que leurs dimensions dépendent de la taille des menus, de la nature des messages à construire, de la dimension de l'écran et des aptitudes de l'utilisateur.

La figure 1.8 présente quelques configurations possibles de l'écran partagé par une fenêtre de menu, deux fenêtres de message et une fenêtre de renseignement.

I.8 Accélération de la construction des messages primaires [TIC. 85] [OUA. 86] [HEC. 83]

Le problème le plus important dans l'aide à la communication pour handicapés est probablement celui de la vitesse extrêmement faible de la communication, due dans la plupart des cas à la lenteur de la construction des messages primaires. Pour palier ce problème, on utilise souvent des techniques spéciales, appelées techniques (ou méthodes) d'accélération, qui permettent d'augmenter la vitesse de la construction des messages primaires, et par conséquent diminuer l'effort nécessaire à la construction.

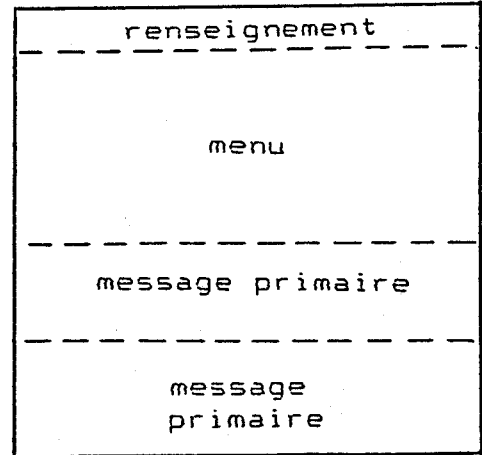
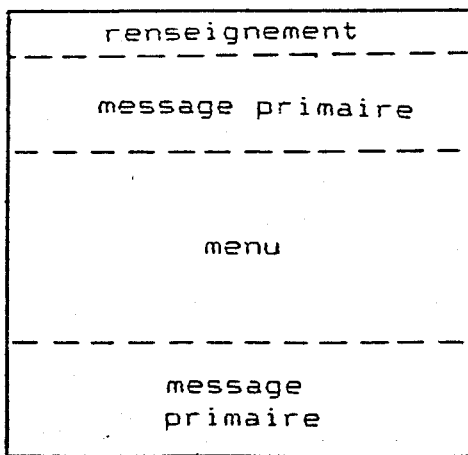
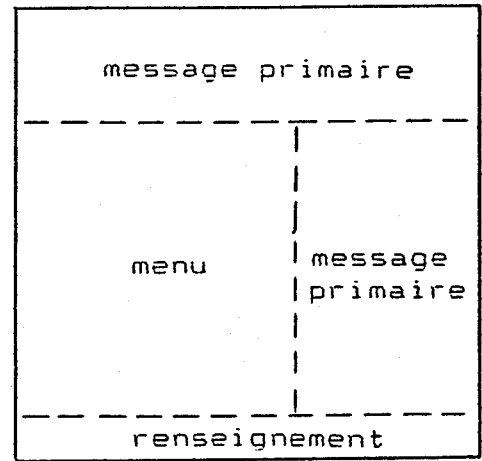
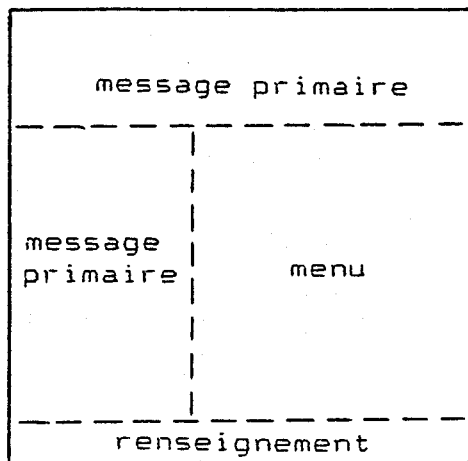
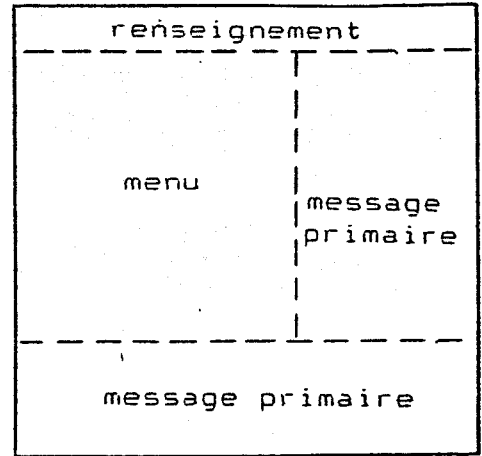
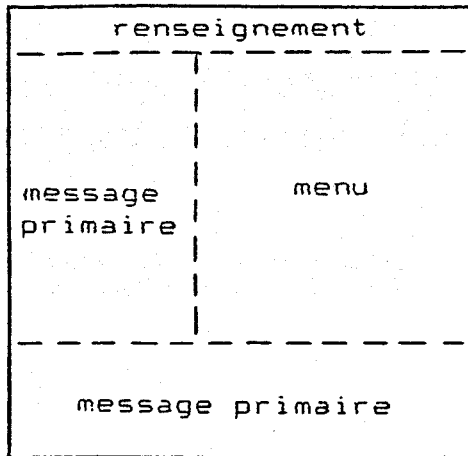


Fig. 1.8 Ecrans de visualisation dans la communication

Le principe de ces techniques consiste à prédire au cours de la communication les éléments primaires à sélectionner ou le message à construire en s'appuyant sur certaines connaissances, soit syntaxiques, soit fréquentielles, soit orthographiques, soit enfin sémantiques ou pragmatiques.

On distingue souvent les cinq méthodes d'accélération suivantes :

- accélération fréquentielle statique,
- accélération fréquentielle dynamique
- accélération par expansion,
- accélération syntaxique et
- accélération sémantique.

I.8.1 Méthode d'accélération fréquentielle statique

La méthode d'accélération fréquentielle statique consiste à présenter les éléments d'un menu dans un ordre correspondant à leur fréquence d'utilisation, et à effectuer la sélection de manière à ce que plus un élément est utilisé, plus sa sélection est rapide.

Par exemple, si les éléments d'un menu représentent les 26 lettres de l'alphabet de la langue française, et si la sélection se fait par balayage pavé par pavé, on peut présenter les lettres dans l'ordre décroissant de leur fréquence d'utilisation, soit :

E A S I T N R U L O D C M P V Q G F B H J X Y Z K W

au lieu de suivre l'ordre alphabétique [MAN. 84].

L'analyse des fréquences d'utilisation des éléments primaires d'un communicateur peut être faite soit avant la réalisation du système, soit de façon adaptative pendant la communication. Dans ce dernier cas, le communicateur enregistre sur une certaine période les fréquences réelles d'utilisation des éléments par son opérateur. Une commande de mise à jour de l'enregistrement, exécutable soit par un moniteur, soit par l'opérateur lui-même, permet d'établir l'ordre de présentation et de sélection des éléments.

La méthode d'accélération fréquentielle statique est la plus simple sur le plan de la réalisation. L'inconvénient est qu'elle ne tient pas compte du message en élaboration.

I.8.2 Méthode d'accélération fréquentielle dynamique

La présentation et la sélection des éléments dans un menu tiennent compte dans ce cas non seulement des fréquences d'utilisation de ceux-ci, mais aussi d'un certain nombre d'éléments sélectionnés. On établit l'ordre de sélection des éléments du menu suivant leur probabilité d'apparition après un ou deux éléments sélectionnés.

A titre d'exemple, dans le cas d'un texte en Français composé à partir des 26 lettres latines, et où on ne tient compte que de la dernière lettre choisie, la présentation des lettres après la sélection de la lettre B peut être la suivante : [MAN. 84]

L R I O A E J S T U B C D F G H K M N P Q V W X Y Z.

Cette méthode permet d'augmenter de façon plus importante la vitesse de la construction des messages primaires, que la méthode d'accélération fréquentielle statistique. Elle est cependant plus difficile à réaliser car l'estimation des probabilités conditionnelles des éléments nécessitent des logiciels spécifiques d'évaluation.

I.8.3 Méthode d'accélération par expansion

Le principe de la méthode d'accélération par expansion consiste à proposer à l'opérateur, après la sélection d'une séquence réduite d'éléments, un mot, une phrase, une longue séquence d'éléments issue d'un dictionnaire.

On trouve ainsi par exemple, l'expansion lettre-mot, lettre-phrase, séquence d'éléments-mot et mot-phrase, ... Une analyse du contexte de la communication est souvent utile pour limiter le nombre de chaînes à proposer après la sélection des premiers éléments. Par exemple, la sélection de la lettre G conduit à proposer tous les mots possibles commençant par G. Mais, si l'on se limite au contexte de la programmation en BASIC, le nombre de mots possibles est réduit à 2. Ces mots sont GOTO et GOSUB [PEC. 84].

Cette méthode exige un dictionnaire dans le système de communication.

I.8.4 Méthode d'accélération syntaxique

Se basant sur la grammaire issue d'une analyse syntaxique des messages primaires, la méthode d'accélération syntaxique consiste à présenter à l'opérateur uniquement les éléments utiles du point de vue grammatical en tenant compte des éléments sélectionnés dans le message en construction.

Par exemple, après la sélection de l'article déterminant le, il est raisonnable de ne visualiser que les noms singuliers masculins du dictionnaire.

Pour réaliser la méthode d'accélération syntaxique, il faut simuler l'ensemble des messages primaires que l'opérateur souhaite construire. On a alors besoin de grouper les éléments primaires et les mots du dictionnaires en classes ou catégories syntaxiques, puis d'établir un ensemble des règles qui assurent la liaison entre ces classes.

Etant donné la difficulté de l'analyse syntaxique sur toutes les structures possibles des messages primaires, on applique souvent la méthode d'accélération syntaxique uniquement aux structures ou "phrases" de messages couramment utilisées par l'opérateur. Cette méthode est particulièrement intéressante lorsqu'il s'agit d'écrire des messages primaires dans un langage informatique.

I.8.5 Méthode d'accélération sémantique

La méthode d'accélération sémantique (ou pragmatique) consiste à présenter à l'opérateur, en tenant compte du sens des éléments sélectionnés dans le message en cours d'élaboration, uniquement les éléments utiles d'un point de vue sémantique. Elle complète la méthode d'accélération syntaxique.

Par exemple, la sélection de l'élément verbe boire doit entraîner la visualisation des éléments représentant le nom des boissons.

Cette méthode est très difficile à réaliser, à cause de la très grande difficulté d'établir des relations sémantiques entre les différents éléments d'un message. Elle n'est applicable que dans des cas particuliers.

I.9 Transformation des messages [TOU. 83] [TIC. 85]

I.9.1 Messages primaires

Un message primaire est la composition des éléments informatifs de base sélectionnés et mémorisés dans un tampon de message.

Le message primaire est conçu et réalisé dans un langage compris de l'opérateur du communicateur. Généralement, il n'est pas compréhensible par son destinataire (humain ou machine). Par exemple, le BLISS est un langage idéographique utilisé par certains handicapés. Il n'est pas compréhensible pour un non initié.

I.9.2 Messages secondaires

Un message secondaire est un message primaire transformé. Il peut avoir perdu sa forme originale, par conséquent éventuellement ne plus être compréhensible par l'opérateur. Il est directement utilisable par son destinataire.

I.9.3 Transformation des messages primaires en messages secondaires

Il est souvent nécessaire de transformer un message primaire en message secondaire avant de l'envoyer vers son destinataire pour qu'il puisse lui être compréhensible.

Les transformations les plus utilisées peuvent être les suivantes :

- Français simplifié --> Français,
- Français condensé --> Français,
- Français --> phonèmes,
- Français --> Braille (pour un destinataire non voyant)

- BLISS --> Français,
- PAR-LE-SI-LA-B --> Français,
- mnémoniques --> signaux de commande de dispositifs périphériques, ...

Il est possible que, dans un message primaire, des erreurs soient détectées par la procédure de transformation. Dans ce cas, le système de communication doit être capable de signaler les erreurs et de retourner au menu correspondant avec lequel l'opérateur peut corriger les erreurs.

I.10 Emission des messages secondaires

Le communicateur passe un message secondaire à l'environnement (interaction communicateur-environnement) en l'émettant vers le dispositif destinataire par l'interface de sortie correspondante.

Les messages secondaires que l'opérateur souhaite émettre à l'environnement peuvent se classer en deux catégories :

- les informations exprimées dans une langue usuelle ou pictographique, qui sont à émettre vers des dispositifs de sortie tels que :

- écran d'affichage,
- imprimante,
- synthétiseur vocal,
- ordinateur [NEL. 83],
- afficheur braille [GRO. 86], ...

- les signaux de commande des dispositifs comme :

- bras-robot,
- fauteuil roulant électrique,
- conditionneur d'air,
- éclairage,
- téléviseur et radio, ...

Il est utile de permettre à l'opérateur de répéter en cas de nécessité l'émission d'un message par une simple commande.

Conclusion

Le système d'aide à la communication pour handicapés est un dispositif servant à augmenter la capacité de communication de son opérateur handicapé.

Il permet à son opérateur de sélectionner et d'assembler des constituants (éléments) primaires prédéfinis dans le système, à l'aide d'un organe d'accès et d'une méthode de sélection, pour construire des messages primaires. Ces messages sont alors mémorisés dans le système, visualisés sur un écran vidéo, éventuellement transformés afin d'être compréhensibles et envoyés vers les destinataires correspondants.

L'organe d'accès et la méthode de sélection doivent être complètement adaptés aux besoins et à la capacité physique et mentale de l'opérateur.

Des techniques d'accélération peuvent être intégrées dans le système. Elles permettent d'augmenter la vitesse de la construction des messages primaires.

L'étude faite dans cette partie du mémoire nous permet de généraliser le fonctionnement du système d'aide à la communication pour handicapés, et d'en dégager alors une structure universelle du système, tant sur le plan du matériel que sur le plan du logiciel. Cette structure constitue le noyau du système de communication, et elle fait l'objet du prochain chapitre du mémoire.

CHAPITRE II :

**CONCEPTION DE LA STRUCTURE D'UN SYSTEME GENERAL
INDIVIDUALISABLE D'AIDE A LA COMMUNICATION
POUR HANDICAPES**

Introduction

Un système de communication est basé sur un micro-ordinateur commercialisé doté d'un logiciel de communication. Ce dernier est composé d'une partie de données et d'une partie de programmes. La première partie, plus importante et qui varie beaucoup d'un système à un autre, comprend les données de base (pavés et menus), peut se structurer de façon identique pour tous les communicateurs. La deuxième, qui représente les programmes exécutables du logiciel, peut être généralisée par une procédure principale de communication et des fonctions modulaires changeables en fonction des besoins de chaque système.

Dans ce chapitre, nous présentons d'abord la configuration du matériel et du logiciel de base d'un système de communication, ensuite nous décrivons de manière détaillée la conception du logiciel général de communication. Les matières à traiter sont les suivantes :

- la structuration des données de base utilisées dans la communication (pavés et menus) par la définition d'une structure générale et "universelle" de ces données;
- la généralisation de la procédure principale du programme de communication par une suite répétitive d'opérations : sélection d'un pavé, interprétation du pavé sélectionné et exécution de la fonction associée au pavé;
- la conception des modules fonctionnels pour les méthodes de sélection de pavés, la mémorisation et la visualisation des messages primaires, l'intégration de procédures de transformation et d'accélération, ainsi que des éléments de commande spéciaux.

II.1 Choix du matériel et du logiciel de base pour le système de communication

II.1.1 Choix du matériel de base

Le concepteur d'un système d'aide à la communication pour handicapés ne peut choisir à son gré le ou les organes d'accès. Ceux-ci sont en effet imposés par la capacité physique de l'opérateur handicapé. Il en va de même pour les organes de sortie, qui sont déterminés par l'environnement de la communication. Ces organes sont tellement variés qu'il n'est possible de prévoir pour un système général que des interfaces d'E/S standards supposées connectables avec ces organes.

Il reste donc à choisir le noyau du système concernant le traitement d'informations. Un micro-ordinateur du commerce satisfait toutes les exigences du système d'aide à la communication [AYL. 81], soit :

- la saisie des informations en provenance de l'organe d'accès grâce à ses interfaces d'E/S standards,
- le traitement des informations,
- la mémorisation temporaire des informations en mémoire vive, et si nécessaire la sauvegarde sur disquette,
- un moyen facile de visualisation des informations grâce à son écran vidéo,
- et la sortie d'informations vers l'environnement par les interfaces d'E/S connectables directement avec la plupart des dispositifs.

Il permet de plus :

- la possibilité de mise en ROM du programme de communication,
- la facilité d'extension et de modification,
- la possibilité d'utilisation d'un téléviseur ordinaire comme dispositif de visualisation,
- et un prix accessible par des handicapés ayant de faibles ressources financières.

La conception générale du système est indépendante de la machine cible. Nous n'avons fixé ni la marque ni le type du micro-ordinateur. Le choix devra être fait en fonction des besoins et de la capacité financière de l'opérateur au moment de la particularisation du système.

II.1.2 Choix du langage de programmation

Nous avons choisi le langage Forth comme langage de programmation pour notre système. Les motifs du choix peuvent s'exprimer par les faits suivants : [BRO. 81] [JAF. 85]

- Le langage Forth existe pour presque tous les micro-ordinateurs commercialisés, sinon, il est facilement implémentable;
- C'est un langage évolué, défini suivant un standard, ce qui permet une grande portabilité;
- Il est structuré et interactif, ce qui facilite le développement et la modification des programmes;
- Il consomme très peu de mémoires par rapport aux autres langages évolués;
- Il est facilement ROMable, et permet donc de mettre dans une cartouche d'EPRM le noyau du Forth et les programmes de communication;
- Il permet l'accès direct aux interfaces d'E/S de la machine, et donc l'interaction de la machine avec les organes d'accès et de sortie du système de communication et il est facile à programmer.

II.2 Structuration des constituants primaires

La base d'un logiciel d'aide à la communication est un ensemble de constituants primaires, appelés pavés, organisés en groupes et sous-groupes structurés, appelés menus et sous-menus.

II.2.1 Structure du constituant primaire

Les constituants primaires, i.e. les pavés, intégrés dans un système d'aide à la communication sont tous structurés de façon identique. Autrement dit, il existe un type générique de données, le type PAVE, qui définit non seulement la structure interne de pavé, mais aussi la façon dont on accède aux pavés et les opérations que l'on peut effectuer sur les pavés. Chaque pavé est une instance du type générique PAVE.

Un pavé est présenté en mémoire par un bloc de paramètres caractérisant sa représentation visuelle, sa dimension et sa fonction. Il est identifié par sa clé, i.e. l'adresse relative de son bloc de paramètres. Tout accès au pavé doit passer par cette adresse. La somme de cette adresse et l'adresse de référence donne l'adresse du pavé.

La figure 2.1 montre la structure du pavé avec les paramètres et leurs emplacements dans le bloc de paramètres. On y trouve dans l'ordre :

- le code opératoire du pavé (2 octets),
- son mode de description (1 octet),
- son paramètre "propriété spéciale d'affichage" (1 octet),
- ses paramètres "dimension", i.e. les nombres de lignes et de colonnes (2x1 octets),
- et ses codes de description.

- Code opératoire

Le paramètre "code opératoire" d'un pavé représente la fonction ou le rôle du pavé dans la communication.

La fonction d'un pavé peut être l'une des fonctions suivantes :

- pavé affichable

Un pavé affichable représente un élément informatif de base. Son rôle essentiel consiste à être un constituant de message primaire. Certains pavés affichables jouent un rôle supplémentaire qui donne le classement suivant :

- pavé affichable ordinaire,
- pavé générateur,
- pavé modulateur,
- pavé générateur-modulateur,
- pavé vide,
- et pavé affichable-exécutable;

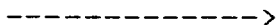
- pavé exécutable

Un pavé exécutable représente une commande intégrée dans le système et utilisable par l'opérateur. Celle-ci est représentée par un seul code, qui peut être en Forth le CFA d'une procédure. Le CFA, ou Code Field Address en Anglais, d'une procédure est en Forth une adresse spéciale dans la définition de la procédure [BRO. 81]. C'est seulement par cette adresse que la procédure est exécutée.

- Mode de description

Le paramètre "mode de description" d'un pavé spécifie la façon dont on codifie la représentation visuelle du pavé. Il y a plusieurs modes de codage. Nous utilisons le mode alphanumérique, le mode incrémental et le mode "pattern". Le paramètre prend la valeur 1, 2 ou 3 respectivement pour ces trois modes.

Adresse
du pavé



code opératoire	2 octets
mode de description	1 octet
propriété spéciale d'affichage	1 octet
nombre de colonnes (en caractères/pixels)	1 octet
nombre de lignes (en caractères/pixels)	1 octet
codes de description	



Fig. 2.1 Structure du pavé

- mode alphanumérique

On utilise les codes ASCII pour représenter les pavés alphanumériques composés de caractères d'ASCII. Un exemple est le pavé MANGER de la figure 2.2(1).

- mode incrémental

Il s'agit d'utiliser le codage FREEMAN pour coder des pavés graphiques. Ce mode de codage est favorable pour les pavés graphiques représentant des contours de dessin. La figure 2.2(2) montre le codage du symbole "homme" en BLISS dans ce mode.

- mode "pattern"

Le codage "pattern" consiste à décomposer le pavé en lignes, et coder chacune de ces lignes par un code numérique. Ce mode est destiné à décrire les pavés graphiques représentant des formes complètes de dessin, ou des contours compliqués. La figure 2.2(3) montre un symbole "étoile" codé dans ce mode.

- Propriété spéciale d'affichage

C'est un paramètre supplémentaire à la description qui spécifie des caractéristiques spéciales de visualisation, qui sont hors des codes de description et peuvent être nécessaires à la sélection. On peut trouver :

- inverse-vidéo,
- le soulignage,
- l'encadrement
- les couleurs de fond et de forme.

Par exemple, on peut demander que l'étoile de la figure 2.2(3) soit rouge, et le fond bleu.

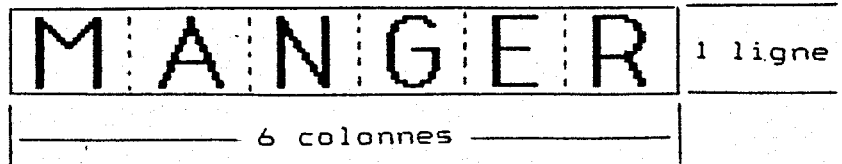
- Paramètres "dimension"

La visualisation des pavés est toujours rectangulaire ou carrée. La dimension d'un pavé est spécifié dans le bloc de paramètres du pavé par le nombre de lignes et le nombre de colonnes. Ces grandeurs sont mesurées soit en caractères pour les pavés alphanumériques, soit en pixels pour les pavés graphiques.

Par exemple, dans les figures 2.2(1) et (2), le pavé MANGER a un caractère en hauteur et six caractères en largeur, le pavé "homme" est de dimension 16x24 en pixels.

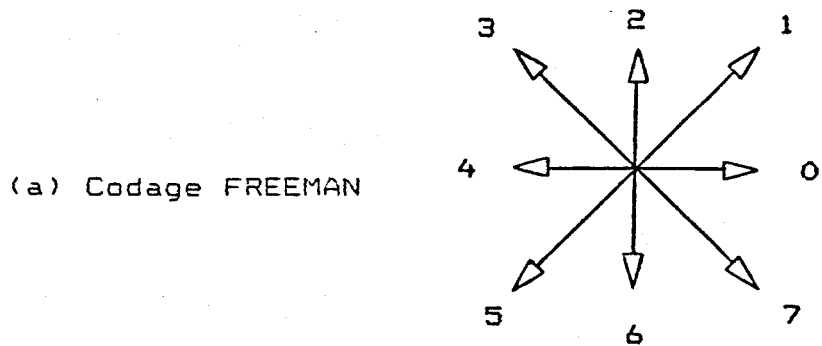
- Codes des description

Les codes de description d'un pavé sont les paramètres qui décrivent le contenu, ou représentation visuelle du pavé. Leur type doit correspondre au mode de description du pavé.

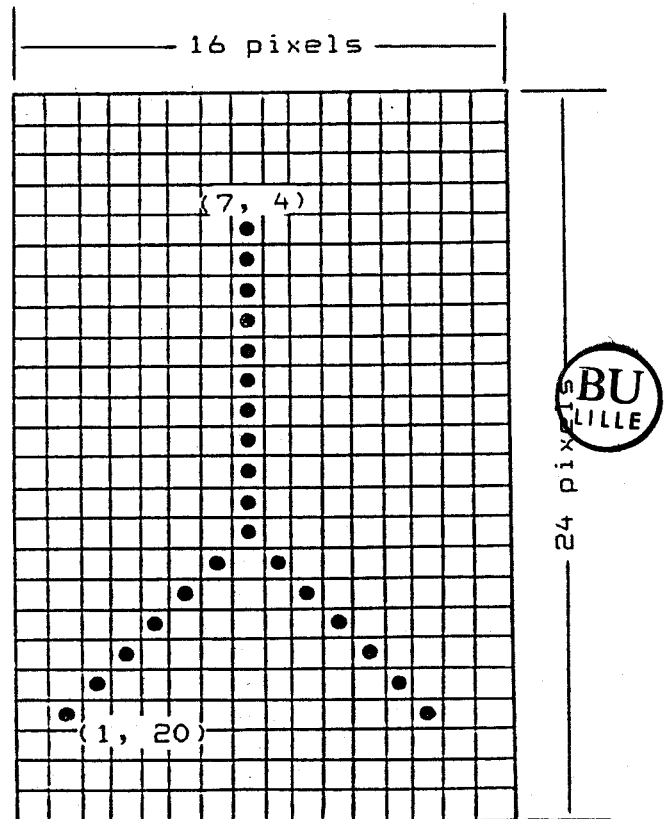


Codes des description: 4D 41 4E 47 45 52

Fig. 2.2(1) Un pavé alphanumérique



(b) Symbole BLISS "homme"



Codes de description:

- 2 : le nombre des branches simples,
- 9 : le nombre des codes pour la 1^{ère} branche,
- 7, 4 : les coordonnées de la 1^{ère} branche,
- 12 : le nombre des codes pour la 2^{ème} branche,
- 1, 20 : les coordonnées de la 2^{ème} branche,
- 6, 6, 6, 6, 6, 6, 6, 6, 6 :
les codes pour la 1^{ère} branche,
- 1, 1, 1, 1, 1, 1, 7, 7, 7, 7, 7, 7 :
les codes pour la 2^{ème} branche.

Fig. 2.2(2) Un pavé incrémental

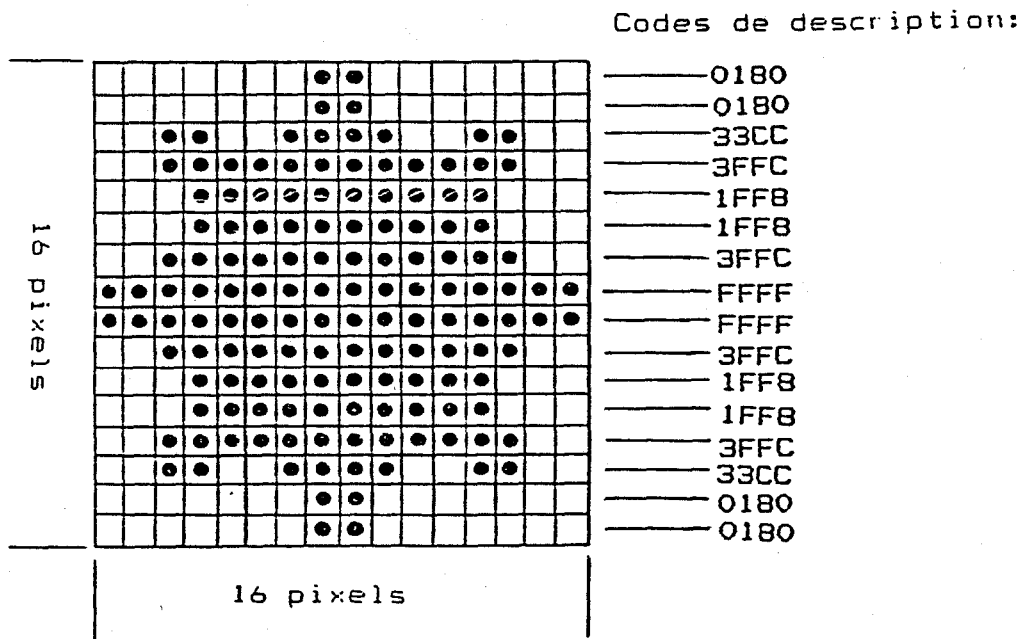


Fig. 2.2(3) Un pavé "pattern" (étoile)

Les codes de description d'un pavé alphanumérique sont une suite de codes ASCII. Par exemple, les codes \$4D, \$41, \$4E, \$47 \$45 et \$52 codent le pavé MANGER .

Les codes de description d'un pavé incrémental sont une suite de chiffres en octal précédés des paramètres spécifiant la décomposition du pavé en branches simples, c'est à dire, plus précisément le nombre de branches simples, le nombre des codes et les coordonnées relatives de chacune des branches. Chaque code octal occupe un demi-octet en mémoire. On trouve dans la figure 2.2(2) les codes de description du pavé "homme".

Les codes de description d'un pavé du type "pattern" est une suite de nombres. Chacun des nombres sert à coder une ligne de pixels. Par exemple, le pavé "étoile" de la figure 2.2(3) est codé à l'aide des 16 codes présentés dans la figure.

II.2.2 Structure du menu

Les constituants primaires sont logiquement organisés en menus et sous-menus. Un menu (ou sous-menu) représente une page visualisable de pavés dans une fenêtre d'affichage. Tous les menus ont la même structure : on peut donc utiliser un type générique de données, dit MENU, qui définit d'une part la structure interne de menu, d'autre part les procédures d'accès et les opérations effectuables sur les menus. Chaque menu est une instance du type générique MENU.

Un menu est représenté en mémoire par un ensemble de paramètres qui occupent un bloc de mémoires consécutives. Ces paramètres définissent les pavés que le menu utilise, la disposition de ces pavés dans une fenêtre d'affichage et la méthode de sélection à l'aide de laquelle l'opérateur peut sélectionner l'un de ces pavés.

La figure 2.3 schématise la structure du menu en mémoire, où les champs sont réservés respectivement pour :

- la méthode de sélection du menu (2 octets),
- son mode de désignation (1 octet),
- son type "texte/graphique" (1 octet),
- la table de pavés où se trouvent les clés des pavés utilisés par le menu et les coordonnées associées aux pavés (nombre des pavés x 6 octets),
- et les paramètres supplémentaires.

- Méthode de sélection

Le paramètre "méthode de sélection" d'un menu définit la méthode de sélection utilisée pour sélectionner l'un des pavés dans le menu. La valeur du paramètre est en effet le CFA de la procédure de sélection correspondante. Les méthodes de sélection implémentées sont les suivantes :

- sélection directe sans codage,
- sélection avec codage ligne-colonne (ou colonne-ligne),
- sélection avec codage dichotomique mixte,
- sélection par balayage pavé par pavé,

Adresse
du menu



Table
de pavés

méthode de sélection	2 octets
mode de désignation	1 octet
type "texte/graphique	1 octet
nombre de pavés	1 octet
X_1	2 octets
Y_1	2 octets
clé du 1 ^{er} pavé	2 octets
. . .	
. . .	
. . .	
. . .	
. . .	
. . .	
X_n	
Y_n	
clé du n ^{ème} pavé	
paramètres supplémentaires	



Fig. 2.3 Structure du menu

- sélection par balayage ligne-colonne (ou colonne-ligne) et
- sélection par balayage dichotomique mixte.

- Mode de désignation

Le paramètre "mode de désignation" complète la spécification de la méthode de sélection du menu. Il indique le moyen visuel de désignation (ou marquage) de pavés pendant la sélection. Il est utilisé par la méthode de sélection pour désigner les pavés sélectionnables à un moment donné. Il prend les valeurs 1, 2, 3, 4, 5 et 6 pour les moyens de désignation respectifs :

- inverse-vidéo,
- clignotement,
- encadrement,
- soulignage,
- flèche-curseur,
- et changement de couleur.

On peut très bien ajouter d'autres modes de désignation en cas de nécessité.

- Type texte/graphique

Le paramètre "type texte/graphique" indique s'il existe des pavés graphiques dans le menu. Il vaut 0 s'il n'en existe pas, sinon 1.

Ce paramètre est utile pour le contrôle de cohérence entre le menu à visualiser et le mode actuel de visualisation, et entre le menu et le tampon de message en cours.

- Table de pavés

La table de pavés est une zone réservée pour les paramètres concernant les pavés utilisés dans le menu et leur disposition dans la fenêtre d'affichage de menu. Elle a autant de cases que le nombre des pavés dans le menu. Ce nombre est conservé dans un octet en tête de la table. Réservee pour l'un des pavés, chaque case de la table contient la clé (l'adresse relative) du pavé et les coordonnées associées. Celles-ci sont par définition les coordonnées du caractère (ou pixel) du coin haut gauche du pavé. Elles sont relatives à la fenêtre d'affichage de menu, et mesurées soit en caractères pour les menus alphanumériques soit en pixels pour les menus graphiques.

L'ordre des pavés dans la table dépend de la disposition des pavés dans la fenêtre d'affichage de menu et de la méthode de sélection associée au menu.

La figure 1.4.I (le paragraphe I.4.3) montre les différentes dispositions des pavés d'un menu dans une fenêtre d'affichage.

Dans le cas de la structure "mosaïque", la sélection par balayage pavé par pavé et la sélection directe sans codage sont permises. Pour la première, les pavés sont rangés dans la table de pavés suivant l'ordre de balayage, et pour la deuxième, suivant l'ordre croissant des codes d'entrée associés aux pavés.

Dans le cas de la structure "ligne", les pavés sont rangés dans la table suivant l'ordre de haut en bas pour les lignes, suivant l'ordre de gauche à droite pour les pavés dans chacune des lignes.

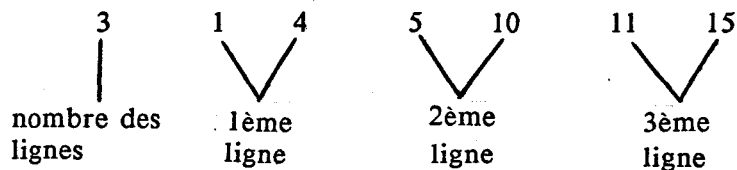
Dans le cas de la structure "colonne", les pavés sont rangés dans la table en colonnes ordonnées de gauche à droite ou de droite à gauche, et selon l'ordre de haut en bas pour les pavés dans chacune des colonnes.

Dans le cas de la structure "matrice", les pavés sont rangés dans la table soit en lignes ordonnées comme pour la structure "ligne", soit en colonnes ordonnées comme pour la structure "colonne". Le choix est dépendant de la méthode de sélection associée au menu.

- Paramètres supplémentaires

Ce sont des paramètres indispensables uniquement dans le cas de la méthode de sélection ligne-colonne (ou colonne-ligne). Ils décrivent explicitement l'organisation des pavés en lignes (ou colonnes) ordonnées. Ils indiquent en effet le nombre de lignes (ou colonnes) de pavés dans le menu, et de façon ordonnée le numéro du premier pavé et celui du dernier de chacune de ces lignes (ou colonnes).

A titre d'exemple, les paramètres suivants décrivent l'organisation des pavés du menu de la figure 2.4 :



1	2	3	4		
5	6	7	8	9	10
11	12	13	14	15	

Fig. 2.4 Exemple de numérotation des pavés dans un menus

II.3 Généralisation de la procédure principale du fonctionnement des communicateurs

En examinant les systèmes existant d'aide à la communication pour handicapés, il est facile de remarquer qu'ils fonctionnent tous suivant une procédure générale identique, quel que soit le système. Cette remarque nous permet de définir un noyau unique pour tous les communicateurs à l'usage des handicapés.

Vue de l'opérateur, la communication assistée revient à effectuer de façon répétitive la même opération : sélection de l'un des pavés présentés à l'aide d'un organe d'accès et d'une méthode de sélection.

Vu du communicateur, le programme de communication consiste essentiellement à poursuivre de façon répétitive les opérations suivantes :

- dans un premier temps, le communicateur acquiert et décode les entrées en provenance de l'opérateur suivant la méthode de sélection utilisée, en vue de la sélection d'un pavé parmi les pavés présentés;

- ensuite, le programme interprète et exécute la fonction du pavé sélectionné. Celle-ci est représentée par le paramètre "code opératoire" du pavé. Son exécution peut conduire :

- soit aux opérations suivantes si le pavé représente un élément informatif de base :

- la mémorisation du pavé dans le tampon de message en cours d'élaboration,

- la visualisation du pavé dans le message, et

- les opérations d'accélération;

- soit à l'exécution de la commande correspondante si le pavé représente une commande exécutable.

Cette procédure générale est représentée par l'organigramme de la figure 2.5

Notons qu'en réalité, l'opération de test après la sélection d'un pavé n'existe pas car il n'y a qu'à exécuter directement le code opératoire du pavé pour exécuter la fonction du pavé, et qu'il existe une étape d'initialisation et une étape de restauration respectivement avant et après la boucle répétitive.

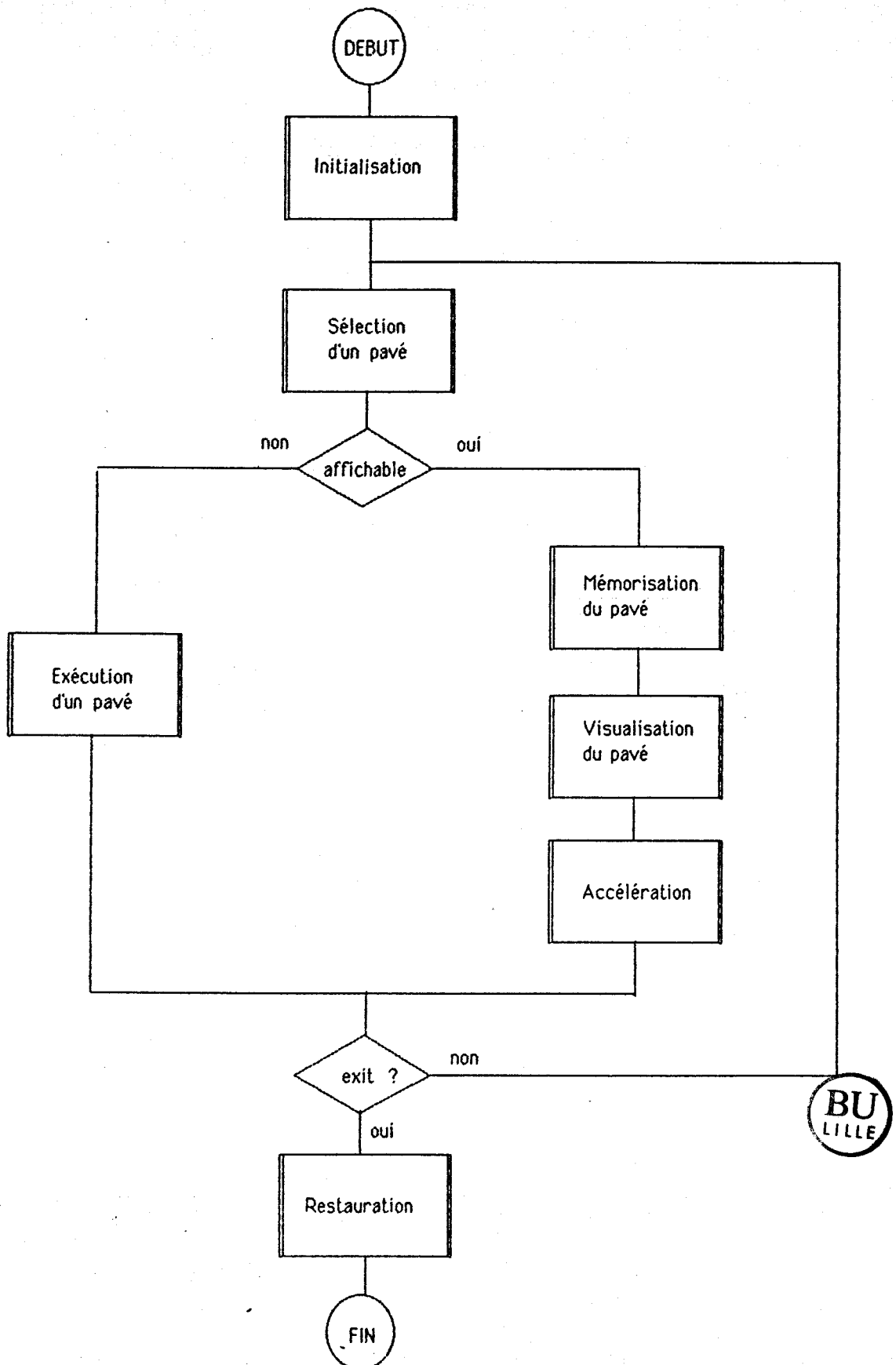
Cette procédure principale de communication garantit une conception adaptée à tous les cas de figure après une particularisation cas par cas.

II.4 Conception des méthodes de sélection

II.4.1 Introduction

La méthode de sélection et la représentation du pavé sélectionnable pendant la sélection sont entièrement déterminées par le menu lui-même. Elles sont en effet spécifiées

Figure 2.5 Procédure principale de la communication



par les paramètres : méthode de sélection et mode de désignation. On ne permet donc pour un menu donné qu'une seule méthode de sélection.

La sélection commence par l'exécution du paramètre "méthode de sélection" pour déclencher la procédure de sélection correspondante. Le paramètre est en effet le CFA de la procédure. Celle-ci saisit et transcode les entrées de l'organe d'accès pour aboutir à la sélection d'un pavé. Elle utilise le paramètre "mode de désignation" pour choisir une procédure appropriée qui effectue le marquage et le démarquage de pavés.

La tâche de saisie et de décodage des entrées est liée au matériel. Elle est confiée à un module qui ne change qu'en cas de changement de machine ou d'organe d'accès.

A la sortie de la procédure de sélection, nous avons comme résultat la clé du pavé sélectionné. Cette dernière est sauvegardée dans une variable.

Quand il s'agit d'une méthode de sélection par balayage, la vitesse de balayage peut être réglée par l'opérateur dans une échelle et un intervalle prédéfinis grâce à un pavé de commande spécial.

La figure 2.6 montre l'organigramme de la procédure générale de sélection.

Nous allons maintenant étudier la conception des diverses méthodes de sélection.

II.4.2 Sélection directe sans codage

Il s'agit d'une méthode de sélection positionnelle. Les organes d'accès utilisables sont le clavier multi-touche et la tablette à digitaliser ou quelque chose de semblable. Les pavés du menu présenté sont rangés dans la table de pavés du menu de manière que la position d'un pavé dans la table corresponde au code entré par la touche ou à la zone utilisée pour positionner le pavé.

La sélection consiste à lire un code de l'organe d'accès, et à indexer la table de pavés avec ce code pour y trouver le pavé positionné. La cohérence de code d'entrée est contrôlée. En cas de code non valide, le système signale l'erreur par une alarme sonore ou visuelle (Figure 2.7).

II.4.3 Sélection avec codage ligne-colonne (ou colonne-ligne)

Les pavés du menu en cours sont rangés dans la table de pavés en lignes (ou colonnes) ordonnées suivant les paramètres supplémentaires. Pour la procédure de sélection, les lignes (ou colonnes) sont numérotées, de même que les pavés dans chacune des lignes (ou colonnes).

La sélection d'un pavé comprend deux étapes :

Figure 2.6 Sélection d'un pavé

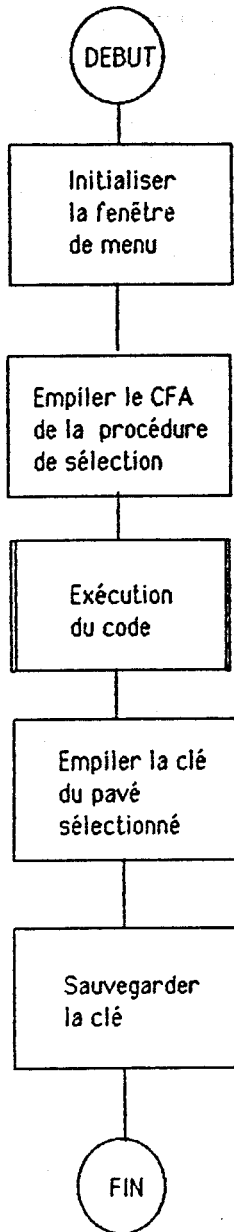
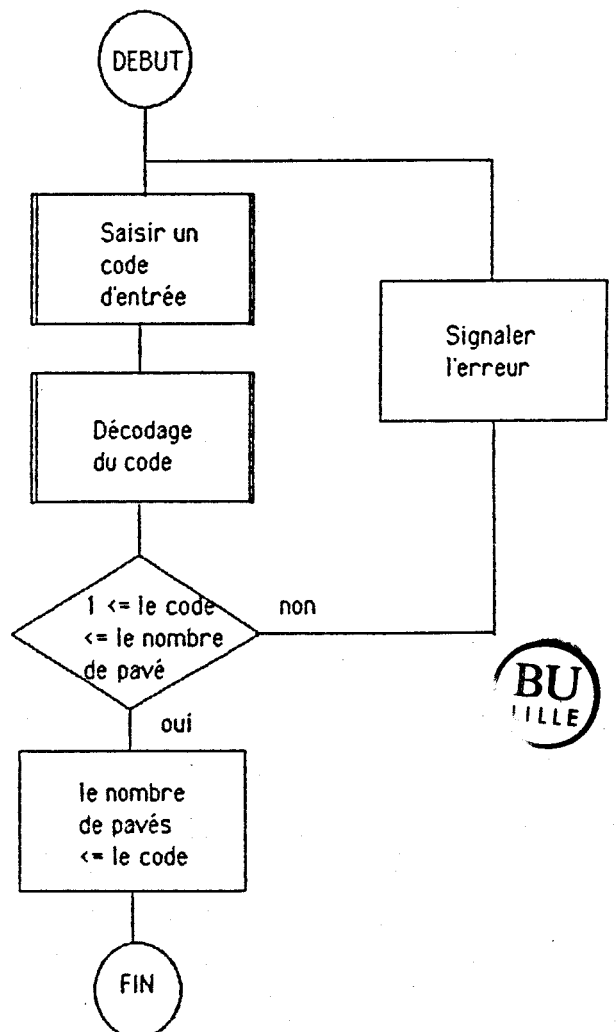


Figure 2.7 Sélection sans codage



- lorsque le système lit et détecte un code d'entrée, il sélectionne une ligne (ou colonne) en la désignant si le code est valide;

- lorsque le système lit et détecte un deuxième code d'entrée, il prend en compte un pavé dans la ligne (ou colonne) sélectionnée, si le code est valide.

Dans le cas où l'un de ces codes n'est pas valide, le système signale l'erreur, et attend un nouveau code d'entrée (Figure 2.8).

Les paramètres supplémentaires sont utilisées par la méthode pour vérifier la validité du code d'entrée et pour trouver la ligne (ou colonne) et le pavé sélectionnés dans la table de pavés.

II.4.4 Sélection avec codage dichotomique mixte

La condition exigée par cette méthode est que le nombre des pavés du menu présenté soit une puissance de deux, i.e. $n = 2^m$ avec n le nombre des pavés et m un entier positif.

Dans la table de pavés du menu, les pavés sont rangés en lignes (ou colonnes) ordonnées. Les paramètres supplémentaires ne sont pas nécessaires.

Les pavés sont implicitement numérotés de 1 à $n (=2^m)$ par leur position dans la table de pavés. Par conséquent, le calcul de dichotomisation de l'ensemble et des sous-ensembles de pavés se fait sur les numéros des pavés au fur et à mesure que la sélection progresse.

A chaque niveau de sélection, le système lit et décode un code d'entrée, et il sélectionne l'une des deux moitié de pavés du groupe en cours si le code prend l'une des valeurs : 1 et 2. Un code d'entrée non valide est signalé. Il est possible d'utiliser un troisième code permettant à l'opérateur de valider ou d'abandonner sa sélection. L'organigramme de cette méthode se trouve dans la figure 2.9.

Une itération est utilisée pour assurer la répétition progressive de la sélection.

II.4.5 Sélection par balayage pavé par pavé

Les pavés sont rangés dans la table de pavés du menu affiché suivant l'ordre de balayage, souvent de gauche à droite et de haut en bas sur l'écran. Les paramètres supplémentaires ne sont pas utilisés par la méthode.

Les pavés sont balayés du premier au dernier suivant leur ordre dans la table de pavés et de façon cyclique. L'organe d'accès est sans cesse testé durant l'intervalle entre deux pas de balayage. Une entrée termine le balayage, et sélectionne le pavé désigné au moment où elle arrive. La figure 2.10 présente l'organigramme correspondant.

Figure 2.8 Sélection avec codage ligne-colonne

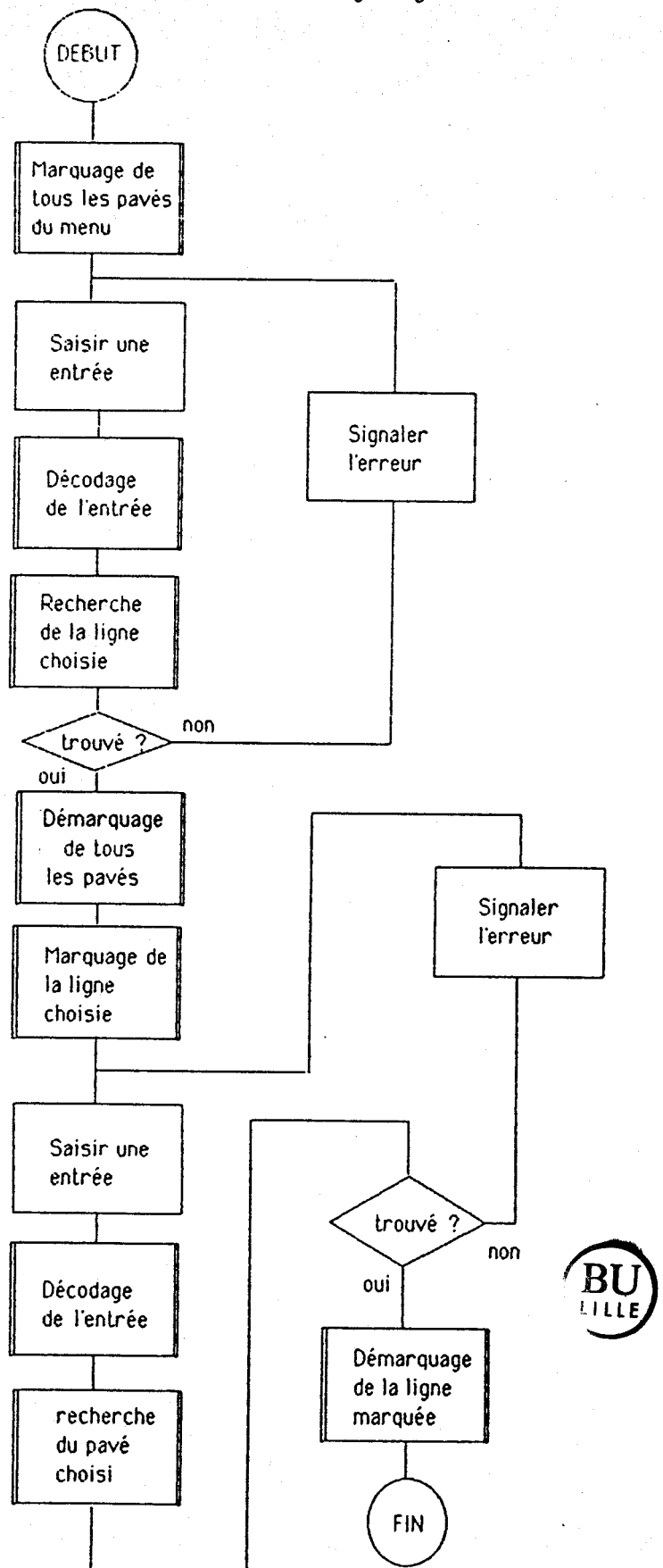


Figure 2.9 Sélection avec codage dichotomique mixte

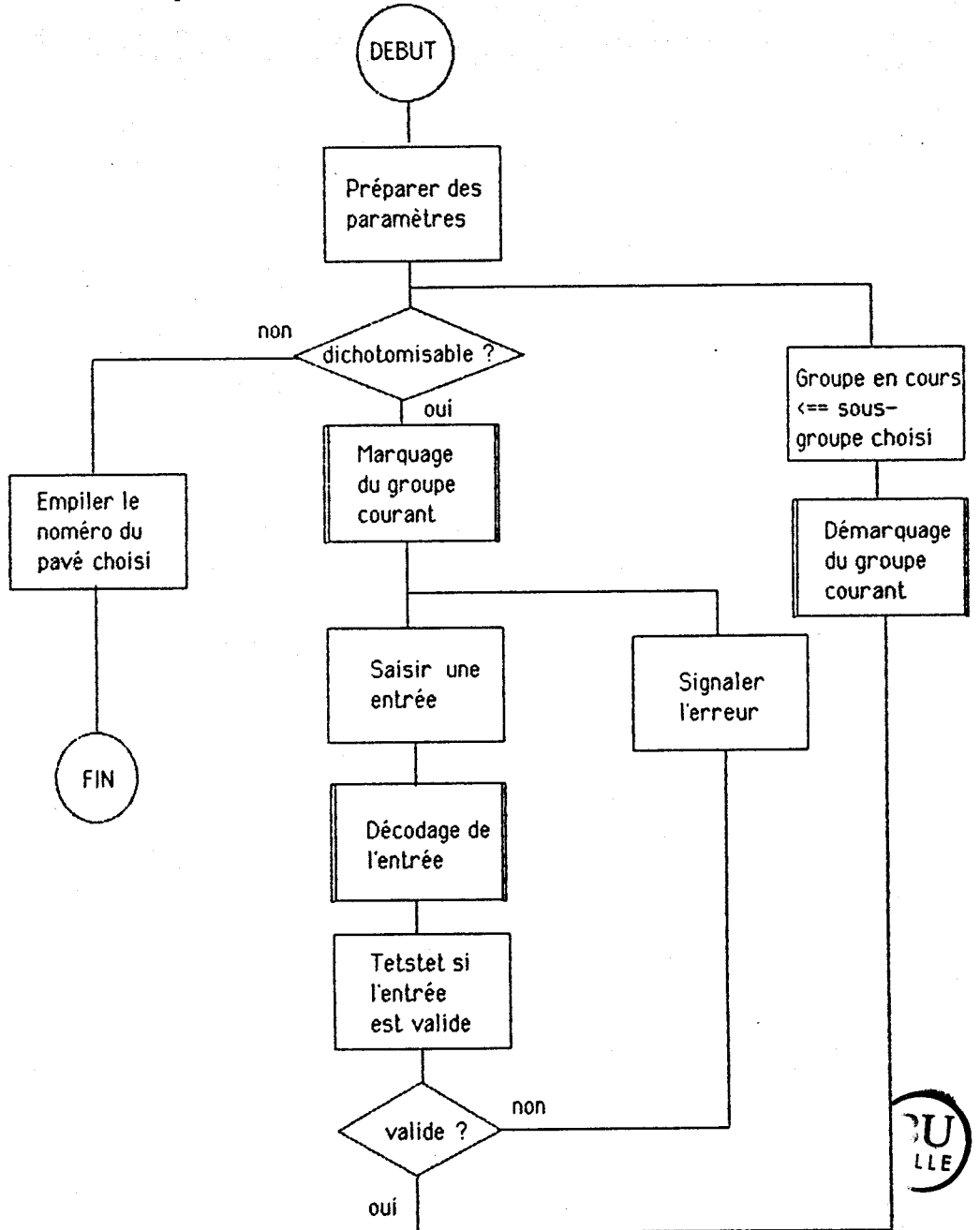
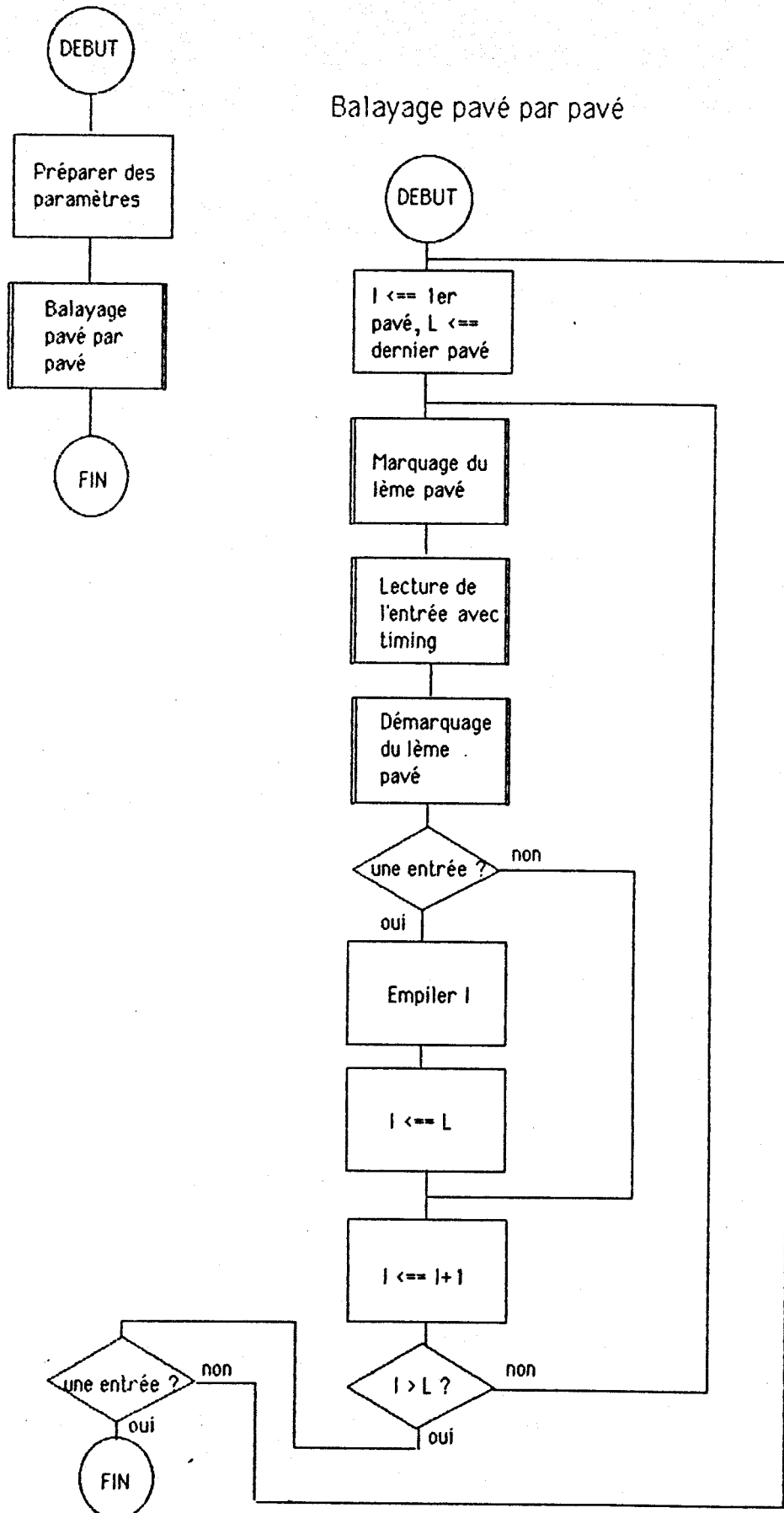


Figure 2.10 Sélection par balayage pavé



La vitesse de balayage est paramétrée par une variable dont la valeur initialement définie par défaut est modifiable par l'opérateur à l'aide d'un élément de commande spécial.

II.4.6 Sélection par balayage ligne-colonne (ou colonne-ligne)

Le rangement des pavés dans la table de pavés du menu affiché est fait de la même manière que pour la sélection avec codage ligne-colonne (le paragraphe II.4.3). Les paramètres supplémentaires sont également nécessaires.

Dans un premier temps, on effectue un balayage ligne (ou colonne), les lignes (ou colonnes) de pavés sont balayées une à la fois de la première à la dernière suivant l'ordre décrit par les paramètres supplémentaires, et de façon cyclique. On teste l'organe d'accès pendant une durée fixée après chaque désignation de ligne (ou colonne). L'arrivée d'une entrée conduit à la sélection de la ligne (ou colonne) désignée. On a comme résultat les numéros des deux pavés délimitant la ligne (ou colonne) sélectionnée.

Ensuite, on effectue un balayage pavé par pavé dans la ligne (ou colonne) sélectionnée en utilisant la procédure conçue pour la méthode de sélection par balayage pavé par pavé. Une entrée sélectionne l'un des pavés de la ligne (ou colonne). L'organigramme de cette méthode est donné à la figure 2.11.

La vitesse du balayage ligne (ou colonne) et celle du balayage pavé sont spécifiées par deux variables et donc séparément réglables par l'opérateur. Pour un opérateur qui a la difficulté de repérage de pavés sur l'écran, le balayage ligne (ou colonne) peut être plus lent que le balayage pavé.

II.4.7 Sélection par balayage dichotomique mixte

Comme pour la sélection avec codage dichotomique mixte, le nombre des pavés du menu affiché doit être une puissance de deux ($n=2^m$) et les pavés sont rangés dans la table de pavés en lignes (ou colonnes) ordonnées. Les paramètres supplémentaires ne sont pas utilisés.

La division dichotomique successive de l'ensemble et des sous-ensembles de pavés se fait au fur et à mesure de la progression du balayage.

A chaque niveau de sélection, on effectue un balayage cyclique sur les deux moitiés du groupe de pavés en cours, tout en examinant l'entrée de l'organe d'accès. Une entrée sélectionne la moitié désignée. On reprend le même balayage sur le sous-groupe sélectionné s'il est encore divisible par deux. La figure 2.12 donne l'organigramme correspondant.

La vitesse de balayage, représentée par une variable, est ajustable par l'opérateur. Il est possible de moduler cette vitesse pour chaque niveau de balayage suivant le nombre des pavés dans le groupe en cours. Plus le nombre est grand, moins le balayage est rapide.

Figure 2.11 Sélection par balayage ligne-colonne(ou colonne-ligne)

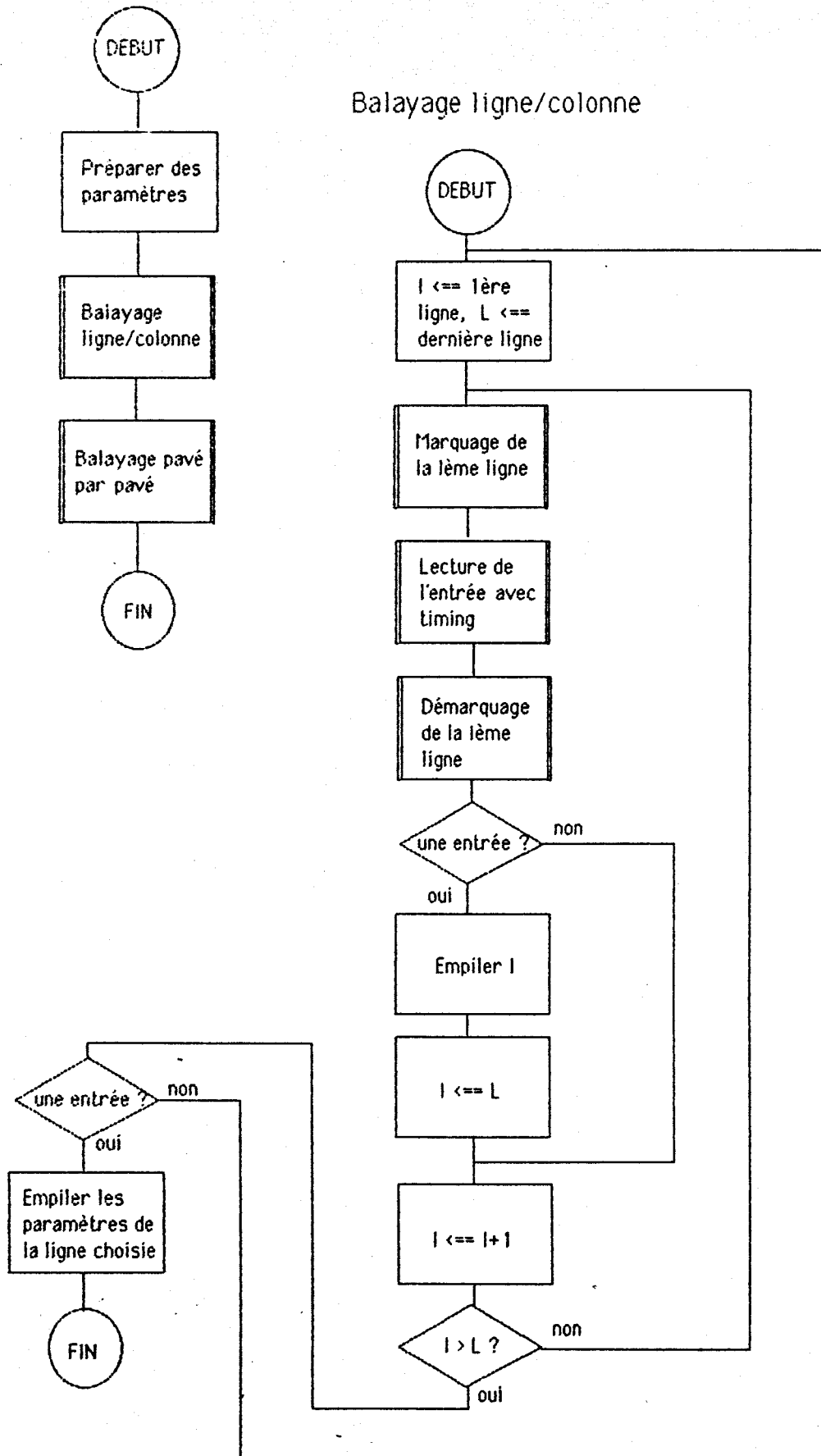
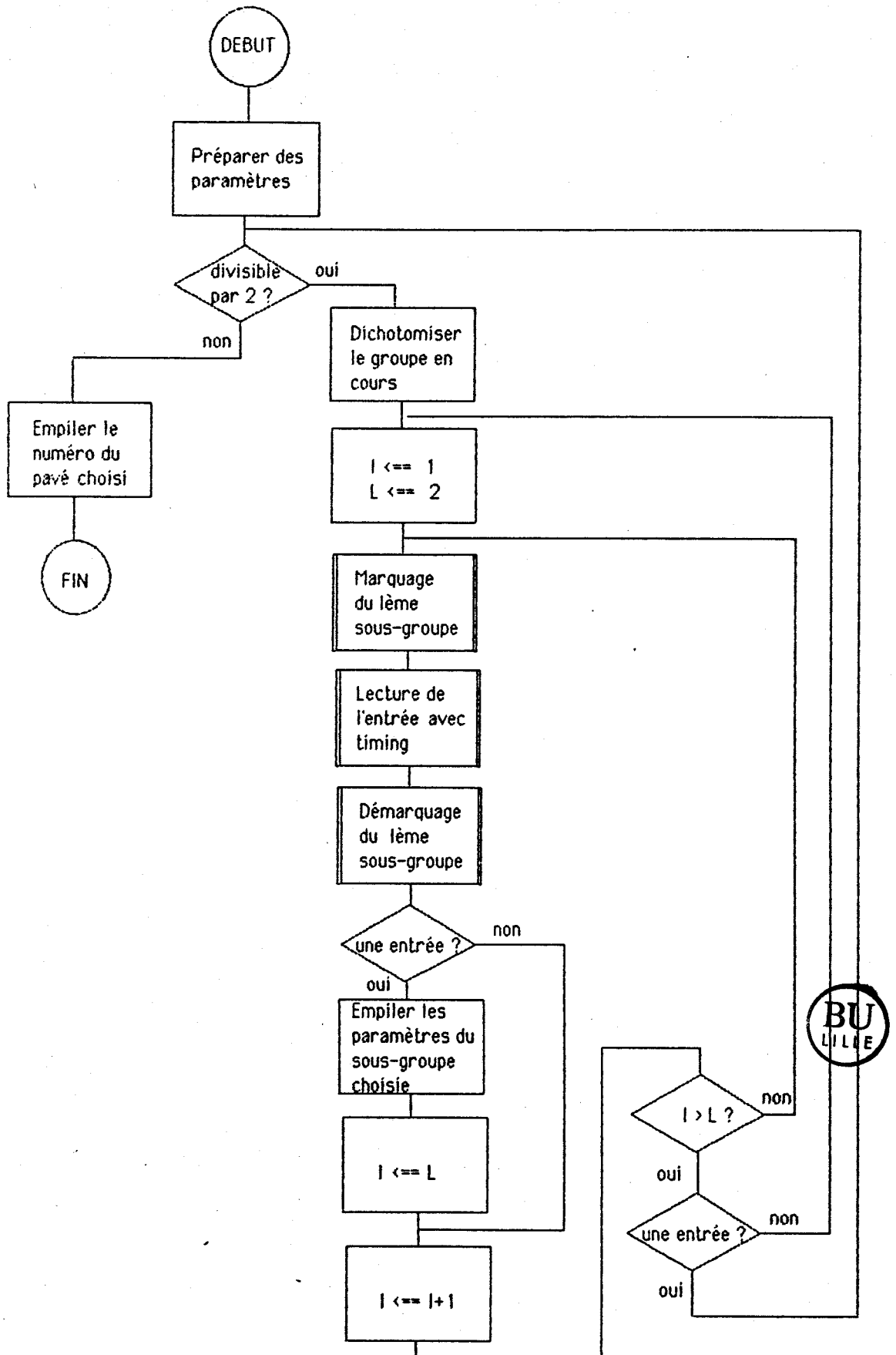


Figure 2.12 Sélection par balayage dichotomique mixte



II.5 Conception de la mémorisation des messages primaires

II.5.1 Structure du tampon de message

Par définition, un message primaire est constitué des éléments informatifs de base sélectionnés. Il est stocké dans un tampon de mémoire, appelé tampon de message ou tampon de mémorisation. Il y est représenté par la suite des clés des éléments sélectionnés accompagnée de certaines autres informations.

Un tampon de message occupe une suite de cases consécutives de mémoire, structurée suivant la figure 2.13. Les différents champs dans la structure sont réservés respectivement pour :

- le compteur (2 octets) des pavés sélectionnés dans le tampon : il contient le nombre des pavés sélectionnés;
- le pointeur (2 octets) du pavé courant : il contient le numéro du pavé sous le curseur;
- le drapeau texte/graphique (1 octet) : il vaut 0 s'il n'y a pas de pavés graphique dans le tampon, et 1 dans le cas contraire;
- le pointeur (2 octets) du tampon précédent : il conserve le numéro ou l'adresse du tampon précédent qui lui est chaîné. Il prend la valeur nulle si le tampon précédent n'existe pas;
- le pointeur (2 octets) du tampon suivant : il contient le numéro ou l'adresse du tampon suivant chaîné, il vaut zéro s'il n'y a pas de tampon suivant;
- les cases (3 octets chacune) réservées aux éléments sélectionnés : chaque élément occupe une case où
 - les deux premiers octets sont utilisés pour conserver la clé de l'élément, ils contiennent initialement la clé d'un pavé vide réel (le paragraphe II.5.3);
 - le troisième octet, appelé indice de propriétés, sert à indiquer certaines propriétés spéciales de l'élément :
 - les deux premiers bits de l'indice, appelés V-bits, sont utilisés pour indiquer si l'élément représente un pavé vide, et ils prennent la valeur 00 si l'élément n'est pas un pavé vide, la valeur 01 s'il est un pavé vide réel, et la valeur 10 s'il est un pavé vide virtuel (le paragraphe II.5.3);
 - le troisième bit, dit M-bit, indique la fonction modulatrice de l'élément, il vaut 1 si l'élément module celui de la case suivante, et 0 dans le contraire;
 - le quatrième bit, appelé O-bit, indique l'origine de l'élément, il vaut 0 si l'élément appartient à l'ensemble des éléments primaires prédéfinis, et 1 s'il est originaire d'un dictionnaire;
 - les autres bits restent libres, ils peuvent servir à d'autres usages, par exemple, pour spécifier le numéro d'un dictionnaire.

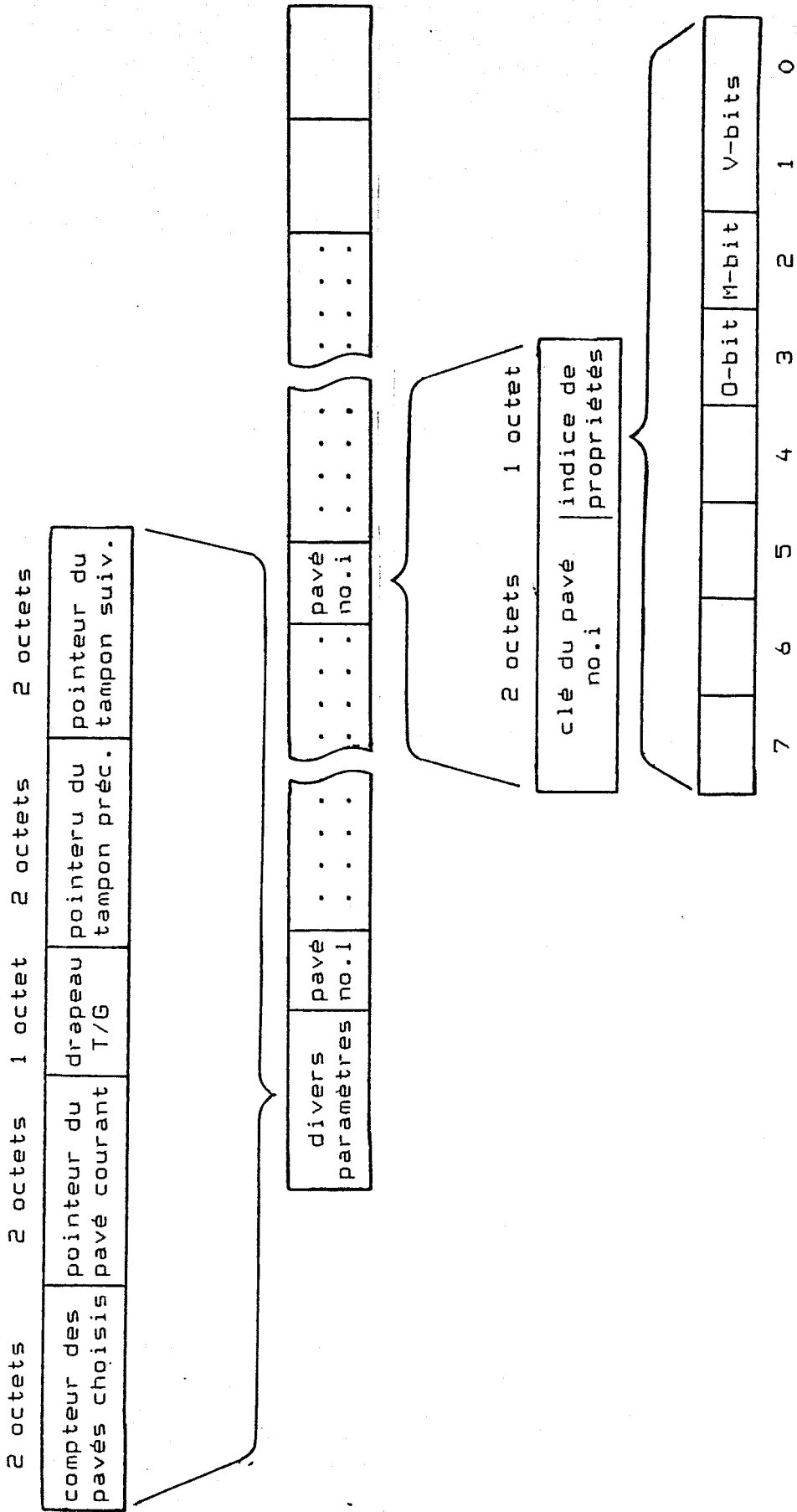


Fig. 2.13 Structure du tampon de message



Le nombre de cases réservées dans un tampon n'est pas fixé par la conception. Il sera déterminé pour chaque système individuel en fonction de la nature des messages à mémoriser et de la quantité de mémoire vive disponible. Il peut exister plusieurs tampons dans un communicateur.

L'accès au tampon de message d'effectue toujours à travers un ensemble de procédures prédéfinies pour cet usage.

II.5.2 Mémorisation des messages primaires

Des tampons de message sont utilisés dans le communicateur pour la mémorisation temporaire des messages primaires. Ces tampons sont numérotés, et mis à la disposition de l'opérateur grâce aux éléments de commande spéciaux permettant de désigner n'importe quel tampon comme tampon de travail, de chaîner et de délier des tampons, de sauvegarder des tampons sur disquette.

Le tampon en cours ou tampon de travail est celui dans lequel les éléments sélectionnés entrent. Il doit être toujours visualisé dans une fenêtre d'affichage.

L'enregistrement d'un élément sélectionné dans le tampon en cours consiste à y mettre la clé de l'élément, et à affecter les autres paramètres concernés. La figure 2.14 montre l'organigramme de la procédure d'enregistrement d'un élément.

La clé de l'élément sélectionné est toujours enregistré dans la case pointée par le pointeur du pavé courant. Si la case contient avant l'enregistrement un pavé vide, ce dernier est remplacé par l'élément sélectionné. Sinon, on effectue soit une insertion, soit un remplacement, en fonction du mode de travail.

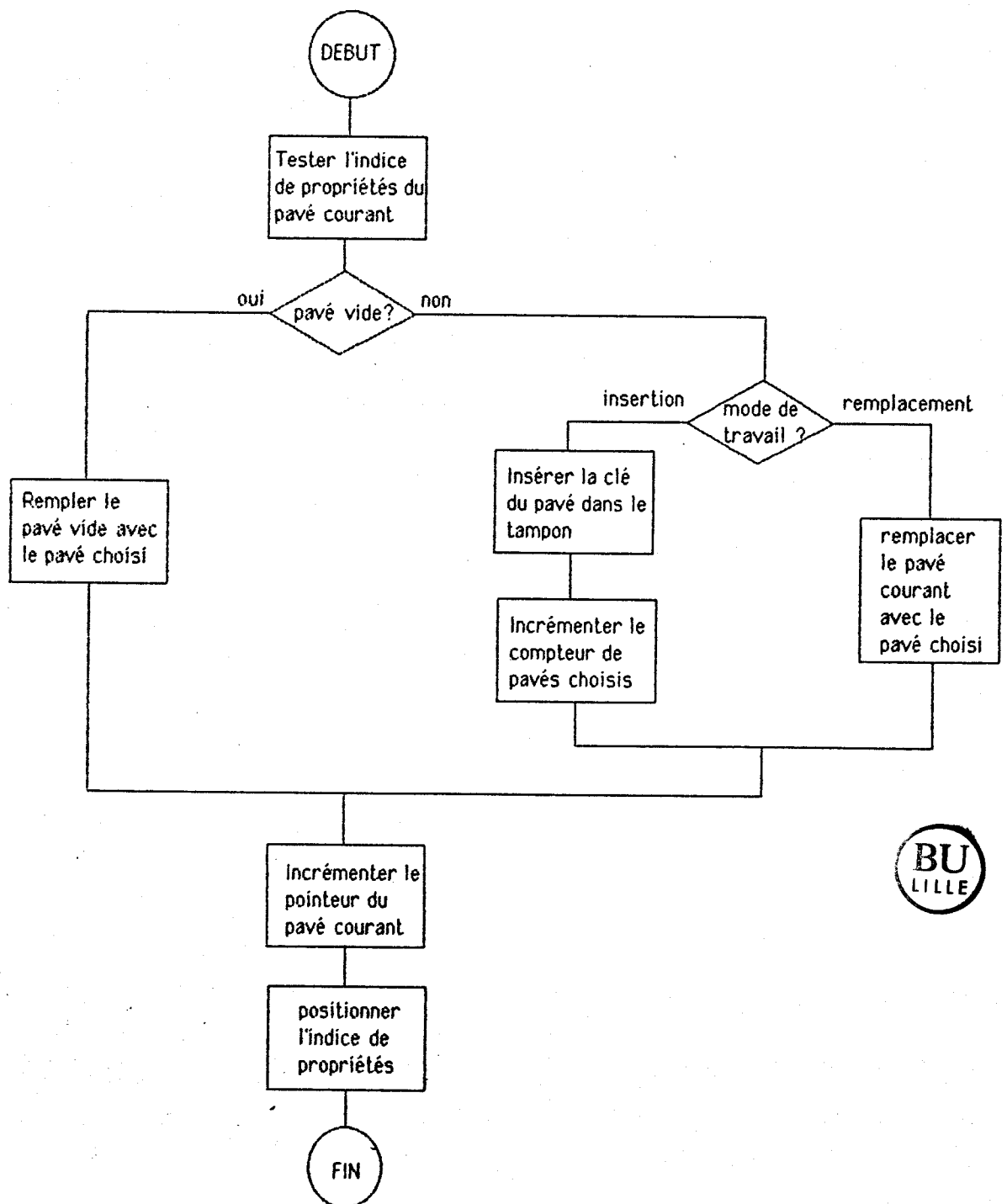
Le débordement du tampon en cours est signalé par le système. Dans ce cas, l'opérateur peut soit changer le tampon en cours, soit chaîner un tampon vide au tampon actuellement en cours.

II.5.3 Conception des pavés vides

Dans le paragraphe I.1.4, nous avons défini, comme pavé vide, un pavé affichable représentant sur l'écran un espace blanc d'une dimension fixe ou variable. La difficulté majeure dans la conception du pavé vide est la variabilité de sa dimension.

Une solution est proposée ci-dessous. Elle pourrait résoudre ce problème de manière simple et efficace. Le principe de la solution consiste à séparer les pavés vides de dimension fixe, des pavés vides de dimension variable, et à réaliser ces derniers sous forme virtuelle à l'aide de la structure de pavé et de la structure de tampon de message.

Figure 2.14 Mémorisation d'un pavé dans le tampon de message en cours



1) Pavés vides de dimension fixe (pavés vides réels)

Un pavé vide de dimension fixe est défini en mémoire avec une structure de pavé normal. Sa représentation visuelle dans le menu qui l'utilise est spécifiée par ses paramètres "mode de description" et "codes de description". Elle peut se composer de symboles spéciaux, pour que l'opérateur puisse le reconnaître. Lors de la sélection du pavé dans un message, il se visualise sous forme d'un espace blanc de la dimension spécifiée par ses paramètres "nombre de lignes" et "nombre de colonnes", grâce à son code opératoire. Ce dernier positionne en plus à la valeur 01 les V-bits de l'indice de propriétés correspondante dans le tampon de message en cours.

Les pavés vides ainsi définis sont sélectionnables par l'opérateur, et destinés à séparer des pavés dans les messages primaires et à initialiser les tampons de message.

Les pavés utiles de ce genre sont souvent le pavé vide "caractère", le pavé vide "ligne de caractères" et le pavé vide "page texte".

Les pavés vides de dimension fixe sont également appelés pavés vides réels, car ils existent physiquement dans le système de communication.

2) Pavés vides de dimension variable (pavés vides virtuelles)

Le but de l'utilisation des pavés vides de dimension variable est de remplacer, lors de la suppression d'un pavé affichable ordinaire d'un message, le pavé par un espace blanc de la même dimension. Cet espace blanc est par la suite traité comme un pavé vide, remplaçable par un autre pavé.

Pour ce faire, nous pouvons définir que si les V-bits de l'indice de propriétés d'une case d'un tampon de message prennent la valeur 10, la case représente un pavé vide virtuel. La dimension du pavé virtuel est déterminée par la clé de pavé qui se trouve dans les deux premiers octets de la case. En effet, lors de l'affichage du pavé virtuel sous forme d'un espace blanc, la clé permet de trouver les paramètres de dimension "nombre de lignes" et "nombre de colonnes" du pavé ordinaire qu'elle représente.

La suppression d'un pavé ordinaire dans un message consiste à effacer la représentation du pavé sur l'écran, et à positionner à 10 les V-bits de l'indice de propriétés de la case correspondante dans le tampon. Elle crée alors un pavé vide virtuel ayant la même dimension que le pavé supprimé.

Cette méthode nous permet d'avoir virtuellement un pavé vide de dimension variable ou d'une autre façon un pavé vide générique de dimension non déterminée et dynamiquement instanciable.

Les pavés vides virtuels n'existent que dans les messages. Par conséquent, ils ne sont pas directement à la disposition de l'opérateur.

II.6 Conception de la visualisation des messages primaires

II.6.1 Multi-fenêtrage de l'écran vidéo [BRA. 86]

Nous utilisons un seul écran vidéo comme dispositif d'affichage pour le système de communication. L'écran est partagé par des fenêtres d'affichage pour visualiser différentes informations. La configuration des fenêtres doit être paramétrée et une fenêtre ne doit pas perturber une autre.

Pour réaliser le partage de l'écran et l'accès généralisé à chacune des fenêtres partageant l'écran, une méthode de multi-fenêtrage a été développée. Elle consiste à définir chaque fenêtre par un bloc de paramètres, et à accéder indifféremment à toutes les fenêtres à travers les procédures d'accès généralisées.

Une fenêtre est une zone rectangulaire de l'écran. Elle est définie par un bloc de paramètres spécifiant sa dimension, sa position, etc (Fig. 2.15). Elle est identifiée par l'adresse de son bloc de paramètres. Les procédures d'accès utilisent cette adresse pour accéder à la fenêtre correspondante.

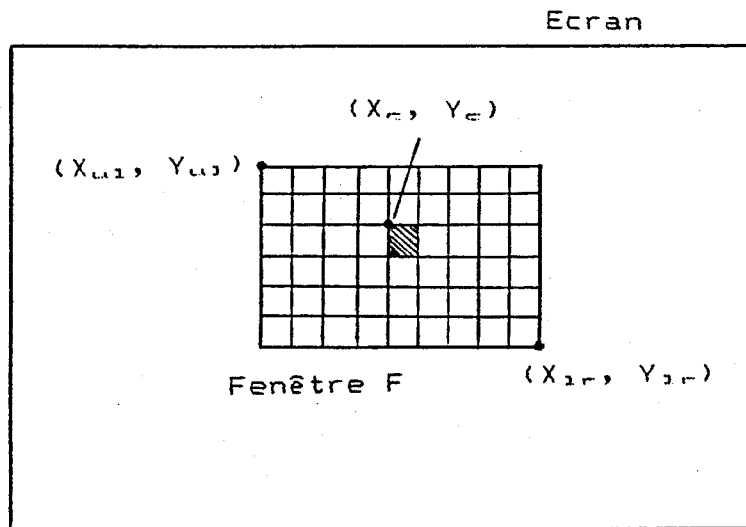
II.6.2 Structure du tampon d'affichage de message

Un tampon d'affichage de message est une zone de mémoire réservée qui contient des informations concernant les éléments affichés du tampon de message. Il lui est associé une fenêtre d'affichage de message. La figure 2.16 présente la structure du tampon d'affichage de message.

Les différents champs dans la structure sont réservés respectivement pour :

- le nombre (2 octets) des pavés visualisés, y compris les pavés vides,
- le numéro (2 octets) du premier pavé visualisé, qui représente la position du pavé dans le tampon de message visualisé,
- le numéro ou l'adresse (2 octets) du tampon de message visualisé,
- l'adresse (2 octets) du bloc de paramètres qui définissent la fenêtre d'affichage associée au tampon d'affichage,
- et les paramètres de chacun des pavés visualisés par le tampon, qui sont :
 - la clé (2 octets) du pavé,
 - l'indice de propriétés du pavé (1 octet),
 - les coordonnées X Y (2x2 octets) sous lesquelles le pavé a été affiché, et qui sont relatives à la fenêtre d'affichage utilisée, et mesurées en caractères si le message ne cours et du type texte, ou en pixels s'il est du type graphique.

Des procédures d'accès à ces paramètres ont été définies. L'accès se fait exclusivement par ces procédures.



La fenêtre F est définie par le sept-uple

$\langle X_{u1} \ Y_{u1} \ X_{1r} \ Y_{1r} \ X_e \ Y_e \ A \rangle$

avec

- $X_{u1} \ Y_{u1}$: les coordonnées du coin haut gauche de la fenêtre,
- $X_{1r} \ Y_{1r}$: les coordonnées du coin droit bas de la fenêtre,
- $X_e \ Y_e$: les pointeurs du curseur de dans la fenêtre,
- A : des attributs d'affichage.



Fig. 2.15 Définition de fenêtre d'affichage

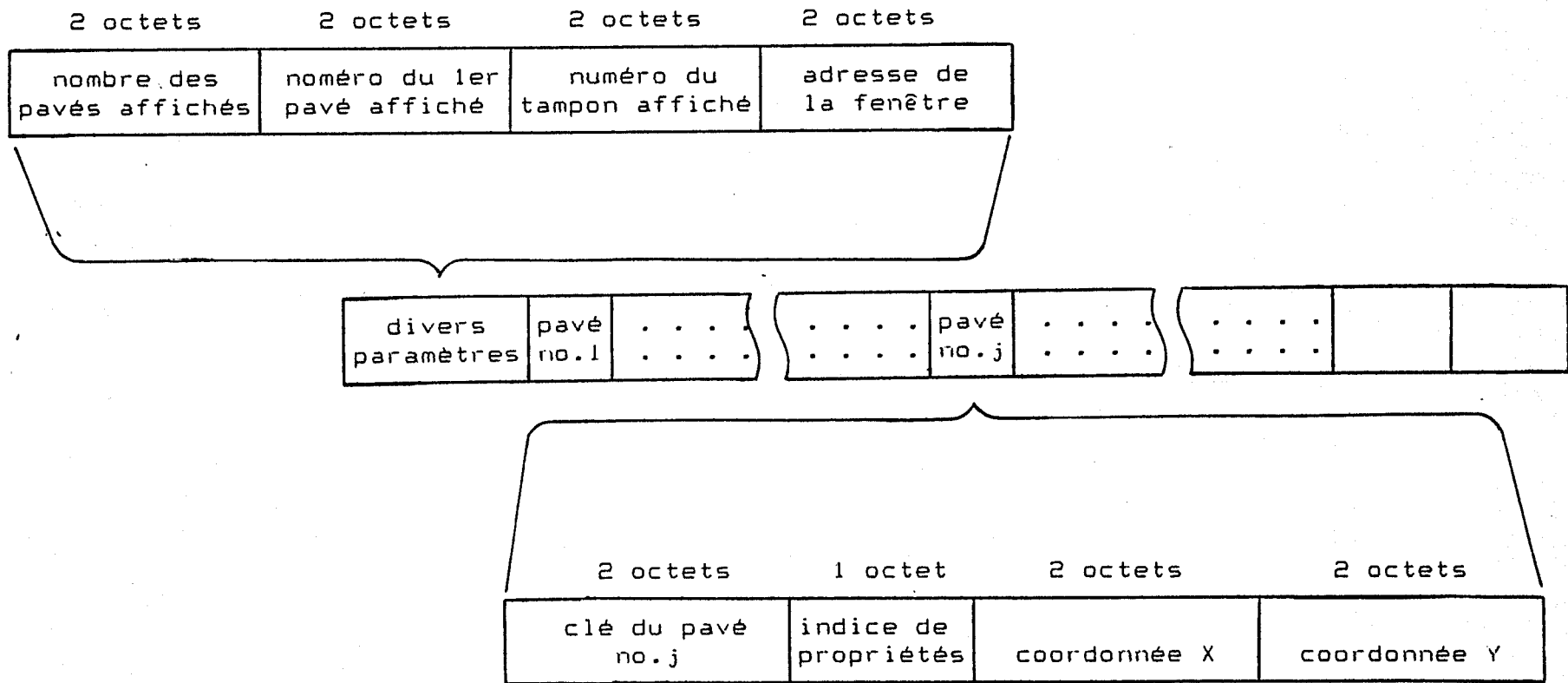


Fig. 2.16 Structure du tampon d'affichage de message

II.6.3 Visualisation des messages primaires

Une ou deux fenêtres d'affichage peuvent être réservées pour visualiser les messages primaires. Dans le cas de deux fenêtres, l'une des fenêtres sert à afficher le tampon de message en cours, et l'autre peut être destinée à visualiser un autre tampon, un tampon d'insertion par exemple. Chaque fenêtre de message est associée à un tampon d'affichage de message.

De façon générale, on ne peut visualiser à un moment donné dans une fenêtre d'affichage qu'une partie d'un tampon de message, i.e. un sous-ensemble parmi l'ensemble des pavés du tampon. Le nombre de pavés dans ce sous-ensemble varie dynamiquement en fonction de la position du sous-ensemble dans l'ensemble, et des dimensions des pavés affichés. En tout cas, il faut que le pavé en cours soit toujours affiché, et que le pavé dernièrement sélectionné se visualise immédiatement après sa sélection.

Quand on affiche un pavé dans une fenêtre de message, on copie sa clé et son indice de propriétés dans un tampon d'affichage correspondant à la fenêtre, et on y enregistre les coordonnées X Y relatives caractérisant la position du pavé dans la fenêtre.

La figure 2.17 montre l'organigramme de la visualisation du message en cours après la sélection d'un nouveau pavé.

Il existe dans le système de communication des éléments de commande spéciaux qui contrôlent la visualisation de message, notamment les commandes de changement de page affichée, les commandes de déplacement de l'indicateur du pavé en cours et les commandes de changement de fenêtre d'affichage.

II.7 Intégration des transformations dans le communicateur

De façon générale, les messages primaires construits ne sont pas directement utilisables par leur destinataire, i.e. l'environnement de la communication. Ils doivent subir une phase de transformation. A l'issue de la phase, ils deviennent messages secondaires compréhensibles par l'environnement.

Etant donné le nombre important de couple de transformation utile "message primaire-message secondaire" dans l'aide à la communication pour handicapés, dont beaucoup sont encore au stade de la recherche, nous ne pouvons pas développer toutes les transformations dans notre système de développement de communicateur. Nous nous sommes contentés de prévoir la possibilité d'interfaçage de procédures de transformation, pour que l'intégration d'une nouvelle procédure de transformation se réalise ultérieurement sans modification de la conception du système.

L'interfaçage des procédures de transformation à la procédure de communication proprement dite peut se situer au niveau des éléments d'émission. Celle-ci vers un dispositif donné appelle automatiquement avant l'émission du message la procédure correspondante de transformation. Ceci est logique car la transformation doit toujours avoir lieu avant l'émission, et l'opérateur n'a pas à contrôler l'usage d'une telle transformation.

Figure 2.17 Visualisation du message en cours
après une nouvelle sélection de pavé

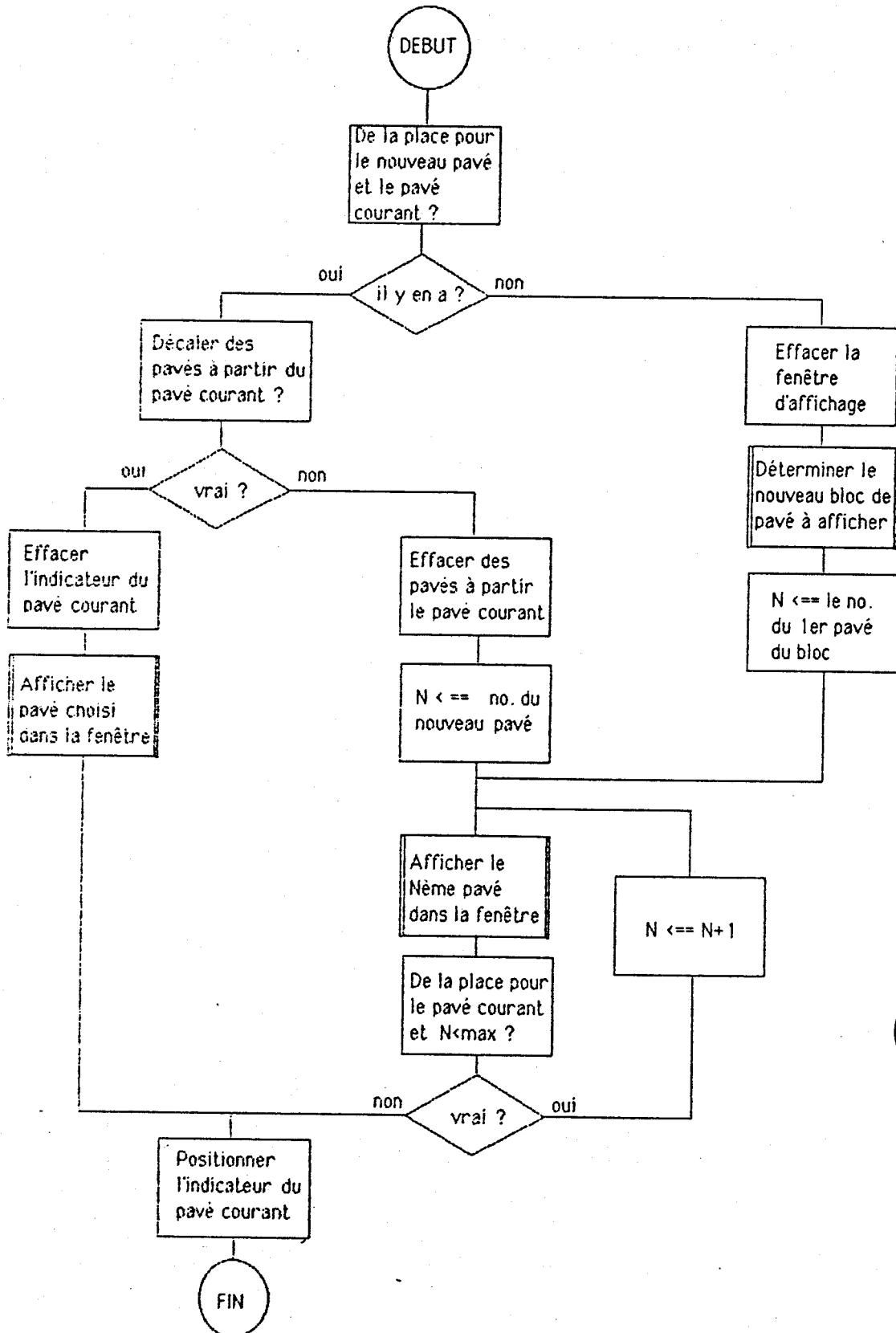
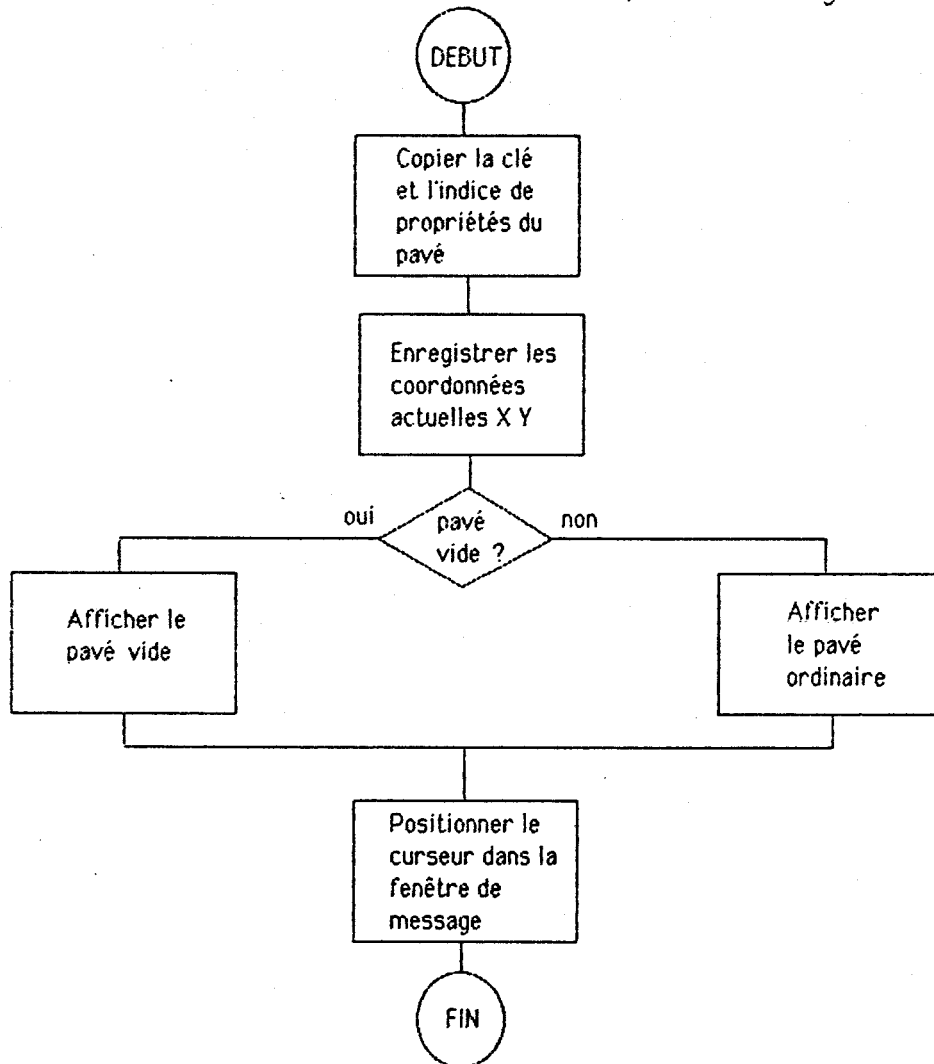


Figure 2.17 (suite) Affichage d'un pavé dans le message en cours avec affectation du tampon d'affichage utilisé



La transformation peut éventuellement repérer et signaler des erreurs dans le message. Dans ce cas, c'est à la commande d'émission de se charger du traitement des erreurs.

Cette approche nous permet d'avoir une conception modulaire modifiable et extensible.

II.8 Intégration des méthodes d'accélération dans le système de communication

II.8.1 Introduction

Parmi les méthodes d'accélération que nous avons étudiées, la méthode d'accélération fréquentielle statique est directement implémentable dans notre système de communication, car elle ne tient pas compte du contexte dynamique de la construction des messages. Pour l'implémenter, il n'y a qu'à ranger les pavés dans les menus suivant l'ordre décroissant de leurs fréquences d'utilisation dans les messages après une analyse statistique préalable sur des textes représentables des messages que peut émettre le handicapé.

Les autres méthodes d'accélération, dont certaines sont encore sujets de recherche, sont moins évidentes à connecter. Nous proposons simplement une solution permettant l'intégration ultérieure d'une méthode quelconque.

Une caractéristique commune des méthodes d'accélération, sauf pour la méthode fréquentielle statique, est qu'elles tiennent toutes compte du contexte de la construction des messages, bien que soient très différentes les façons dont elles procèdent et les connaissances qu'elles exploitent. Ces méthodes consistent de façon générale à proposer des éléments utiles en fonction de certains éléments sélectionnés dans le message en construction. Les éléments ainsi proposés peuvent être :

- les pavés du menu en cours avec ou sans changement de l'ordre de sélection,
- certains pavés d'un autre menu existant,
- ou des éléments d'origines diverses, par exemple issues d'un dictionnaire, formant un menu temporaire paramétré.

Dans tous les cas, il s'agit toujours de proposer un menu.

Cette caractéristique commune nous conduit à proposer la procédure d'accélération décrite par la suite avec les structures des données associées à la procédure.

II.8.2 Structures des données associées à la procédure d'accélération

Les structures des données associées à la procédure d'accélération sont les tampons d'accélération et les tables d'accélération organisées en arbres.

1) Tampon d'accélération

Un tampon d'accélération a une structure de table. Il sert à conserver les clés des pavés sélectionnés sur une certaine longueur dont il faut tenir compte au moment de l'accélération, et les attributs des catégories liés aux pavés, par exemple, leur catégorie syntaxique, leur catégorie sémantique, etc, dans le contexte de la construction du message. Chaque élément du tampon est de trois octets dont deux pour la clé du pavé et un pour les attributs. Un octet en tête de tampon est réservé pour le compteur des pavés dans le tampon. La structure correspondante est montrée figure 2.18(1).

Il peut exister plusieurs tampons d'accélération dans un communicateur.

2) Table d'accélération

Une table d'accélération contient des codes exécutables (des CFA) de procédures qui ont pour mission de visualiser des menus éventuellement après paramétrage et/ou masquage, et les adresses (ou numéros) de ses sous-tables. Il existe souvent plusieurs tables et sous-tables d'accélération dans un communicateur.

Les éléments d'une table sont accessibles par des clés d'accès. Chaque élément contient à part sa clé d'accès (2 octets) un code exécutable (2 octets) et un pointeur de sous-table (2 octets) qui est à zéro si l'élément n'a pas de sous-table. La figure 2.18(2) schématise la structure de la table d'accélération. La longueur de la table est spécifiée par la valeur des deux premiers octets de la table.

Les éléments d'une table d'accélération sont triés suivant l'ordre croissant de leurs clés d'accès pour que l'accès à la table soit rapide.

3) Organisation des tables d'accélération

Les tables et sous-tables d'accélération sont organisées en une structure arborescente. Chaque noeud d'un arbre représente une table ou sous-table, et chaque branche la liaison entre une table "mère" et une table "fille" dans le sens descendant. Cette liaison est représentée par le pointeur de sous-table de l'élément correspondant dans la table "mère". La figure 2.18(3) illustre la structure.

La profondeur de l'arbre d'accélération représente le niveau maximum d'accélération, c'est à dire le nombre maximum de pavés sélectionnés dont il faut tenir compte. Si i est ce nombre à un moment donné, le CFA cherché se trouve alors dans une des tables du $i^{\text{ème}}$ niveau de l'arbre.

Il peut y avoir dans un système de communication plusieurs arbres d'accélération.

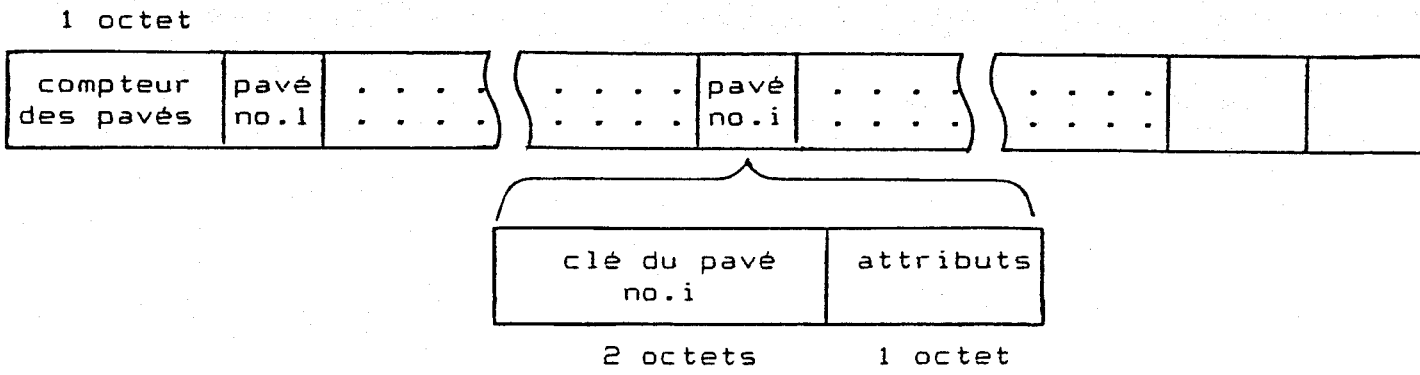


Fig. 2.18(1) Structure du tampon d'accélération

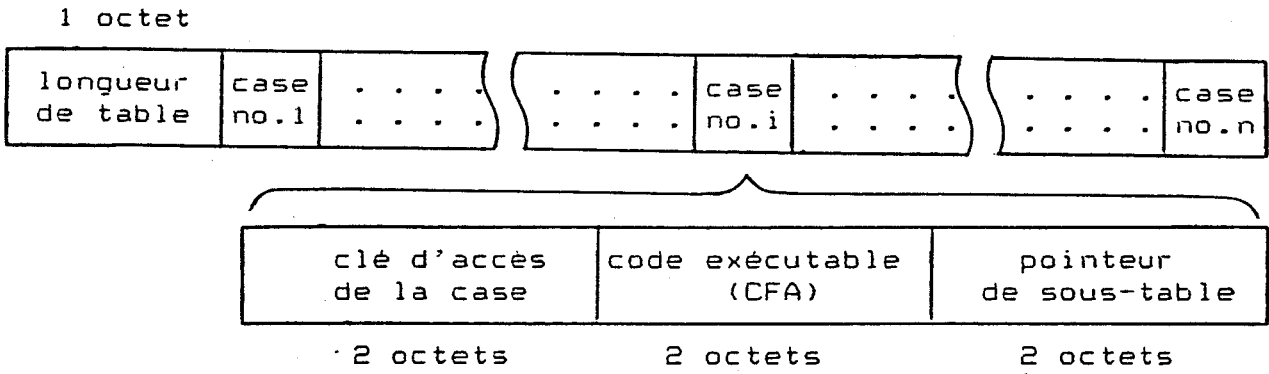


Fig. 2.18(2) Table d'accélération

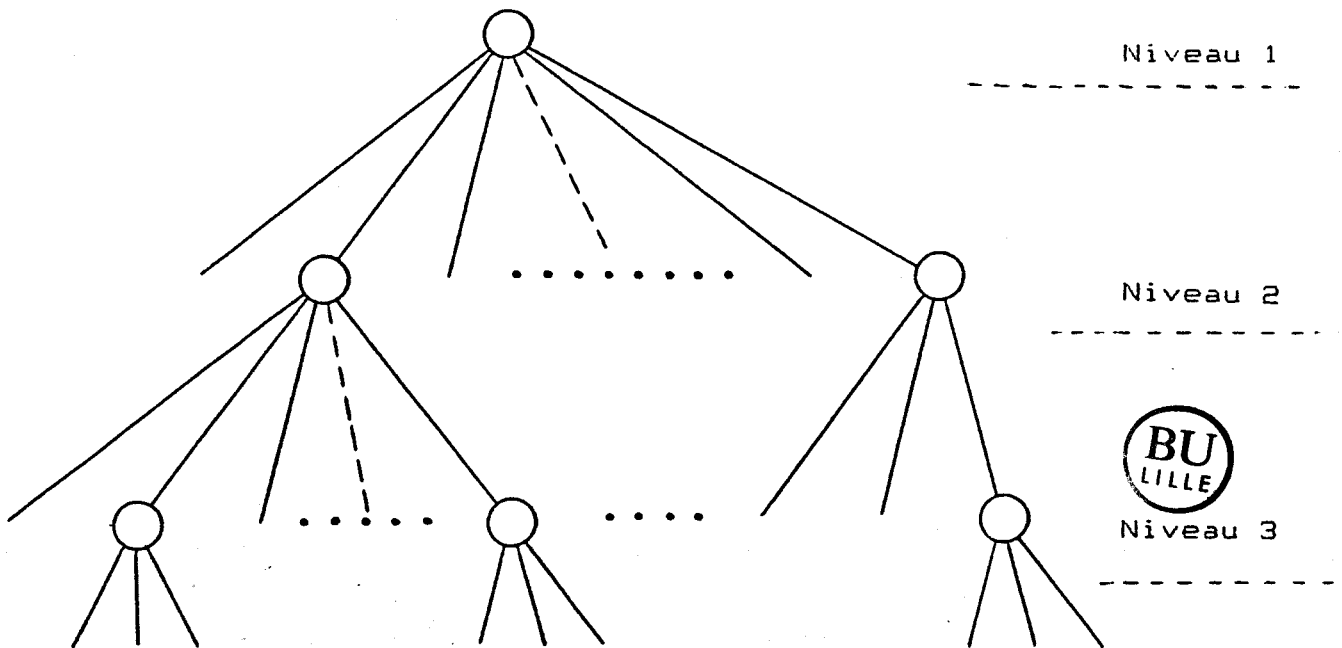


Fig. 2.18(3) Arbre de tables d'accélération

II.8.3 Procédure d'accélération

A l'aide d'un tampon d'accélération et d'un ou plusieurs arbres, la procédure d'accélération consiste après sélection d'un pavé affichable ordinaire successivement :

- à enregistrer la clé du pavé dans le premier élément libre du tampon d'accélération,
- à calculer les clés d'accès avec les données contenues dans le tampon, d'une manière ou d'une autre suivant la méthode d'accélération utilisée ,
- à modifier éventuellement le contenu du tampon,
- à accéder à un arbre d'accélération en utilisant les clés d'accès obtenues précédemment afin d'y trouver le code exécutable correspondant aux clés,
- et finalement à exécuter ce code.

L'organigramme de cette procédure est présentée à la figure 2.19.

La manière de calculer les clés d'accès dépend strictement de la méthode d'accélération utilisée et de la définition des clés d'accès. Dans le cas le plus simple, les clés sont les clés de pavé.

La modification du tampon d'accélération est soit de positionner les attributs de catégories du dernier pavé choisi, soit de ré-initialiser le tampon, par exemple à la fin de la construction du message en cours, etc.

II.9 Conception des éléments de commande

Les éléments de commande sont des pavés exécutables. La sélection d'un tel pavé par l'opérateur entraîne l'exécution de la commande correspondante. Cette dernière est représentée dans la définition du pavé par le paramètre "code opératoire" sous forme du CFA d'une procédure.

Les éléments de commande peuvent se classer suivant leur rôle dans la communication en deux catégories : les éléments de contrôle d'édition des messages et les éléments de contrôle de l'environnement ou d'émission des messages.

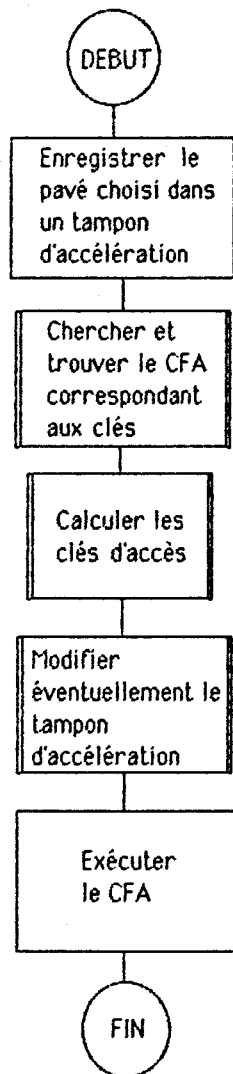
Nous allons étudier ci-dessous de façon non exhaustive la conception des éléments de commande tout en tenant compte des structures de données utilisées pour représenter les éléments primaires, les messages, etc.

II.9.1 Commandes de contrôle d'édition des messages

Les commandes de contrôle d'édition des messages agissent sur les états intérieurs du système de communication.

Les commandes étudiées ci-après peuvent ne pas toutes exister dans un système de communication.

Figure 2.19 Procédure d'accélération après une sélection de pavé



- Déplacement de l'indicateur du pavé en cours

Le pavé en cours d'un tampon de message visualisé, qui doit toujours paraître dans la fenêtre d'affichage utilisée, est visuellement marqué par un curseur spécial appelé indicateur du pavé en cours et réalisé d'une façon ou d'une autre, par inverse-vidéo par exemple.

L'opérateur peut pour désigner le pavé en cours déplacer l'indicateur dans le tampon affiché en avant ou en arrière. Le déplacement peut se faire pavé par pavé ou bloc de pavés par bloc de pavés dans la fenêtre.

- Suppression du pavé en cours

La suppression du pavé en cours consiste à remplacer le pavé en cours par un pavé vide virtuel de la même dimension s'il n'est pas vide, ou à enlever le pavé en décalant les pavés suivants s'il l'est.

Le pavé ordinaire supprimé est conservé par une variable jusqu'à la prochaine suppression pour être éventuellement remis quelque part dans le tampon en cours.

- Insertion de pavé

Le pavé dernièrement supprimé peut s'insérer à la place du pavé en cours. Si celui-ci n'est pas un pavé vide virtuel, l'insertion implique un décalage des pavés suivants. Le système signale le débordement du tampon.

- Changement du tampon de message en cours

L'opérateur peut désigner n'importe quel tampon de message existant comme tampon en cours à l'aide d'une commande spéciale. Celle-ci consiste à proposer la liste des tampons de message existants. L'opérateur doit en choisir un suivant une méthode de sélection, et visualiser le nouveau tampon de message en cours.

- Concaténation de tampons

L'opérateur peut chaîner n'importe quel autre tampon au tampon en cours, et délier le tampon suivant du tampon en cours. Pour le chaînage, la liste des tampons est proposée à l'opérateur afin qu'il puisse en sélectionner un.

Le chaînage est possible grâce aux pointeurs prévus à cet usage dans la structure de tampon de message.

- Changement de la fenêtre de message en cours

Dans le cas où il existe deux fenêtres d'affichage de message, l'opérateur peut décider de visualiser le tampon de message en cours dans n'importe quelle fenêtre à l'aide d'une commande spéciale. La fenêtre d'affichage en cours se distingue de l'autre par un indicateur spécial. Il est possible que les deux fenêtres visualisent le même tampon de message.

La fenêtre en cours est identifiée par une variable dont la valeur est le numéro (ou l'adresse) du tampon d'affichage auquel est associée la fenêtre.

- Visualisation du tampon de message en cours en plein écran

Il est souvent utile de visualiser le tampon de message en cours le plus largement possible. Pour ce faire, une commande spéciale permet de visualiser en plein écran le tampon de message en cours, et éventuellement de faire défiler les pavés dans le tampon.

Une sous-commande existe afin de remettre le système dans son état normal.

- Changement du menu en cours

Il s'agit des commandes permettant à l'opérateur de choisir un menu comme menu en cours. Une telle commande visualise un menu dont le nom est spécifié par la commande. Par exemple, le pavé de commande VERBES affiche un menu contenant des verbes..

- Changement du mode de travail

L'opérateur peut commuter le mode de travail en cours entre le mode d'insertion et le mode de remplacement. En mode d'insertion, un pavé sélectionné s'insère avant le pavé en cours si ce dernier n'est pas un pavé vide virtuel. En mode de remplacement, un pavé sélectionné écrase le pavé en cours quel qu'il soit.

- Initialisation du tampon de message en cours

Une commande permet d'effacer le tampon de message en cours. Il s'agit en effet de remplacer tous les pavés dans le tampon avec des pavés vides réels. La fenêtre d'affichage en cours est elle-aussi initialisée.

- Sauvegarde et appel de tampon de message

Dans le cas où l'accès à une disquette ou cassette est possible, on peut envisager les deux commandes suivantes : l'une sauvegarde le tampon de message en cours dans un fichier, et l'autre effectue le chargement du tampon sauvegardé dans le tampon en cours.

- Ajustement de la vitesse de balayage

Des commandes permettent de régler la vitesse de balayage pour les méthodes de sélection par balayage. La vitesse est initialement définie par défaut, et réglable par l'opérateur sur une échelle et un pas de réglage prédéfinis. Par exemple, pour un balayage pavé par pavé, on a prévu une échelle entre 5 et 30 secondes et un pas de réglage de ± 1 seconde.

Pour la méthode de sélection par balayage ligne-colonne (ou colonne-ligne), la vitesse de balayage ligne (ou colonne) et celle de balayage pavé peuvent être séparément réglables.

Il est intéressant d'avoir une commande permettant de figer la valeur initiale d'une vitesse de balayage avec une valeur réglée. Cette commande exige l'utilisation d'une disquette ou d'une mémoire vive alimentée en permanence.

II.9.2 Commandes de contrôle de l'environnement

Ce sont les commandes qui communiquent d'une façon ou d'une autre avec l'environnement de la communication. Il s'agit pour le communicateur de solliciter un dispositif en lui envoyant des messages et/ou des signaux de commandes via une interface d'E/S adaptée. Ces commandes sont donc appelées commandes d'émission.

Ce qui est envoyé par une commande d'émission peut être un message construit par l'opérateur à partir des éléments informatifs de base et éventuellement transformé, et/ou de divers signaux de commande préprogrammés dans la définition de la commande. Par exemple, la commande, qui permet d'émettre le message contenu dans le tampon en cours vers un synthétiseur vocal, consiste dans un premier temps à traduire en texte le message constitué d'une chaîne de clés de pavés par substitution des clés par les codes de description des pavés respectifs, puis à émettre, vers l'interface d'E/S correspondante, le texte et certains signaux de contrôle spéciaux, par exemple, le signal d'initialisation du synthétiseur et le signal d'indication de fin de texte.

Pour que soit portable la conception de ces commandes, qui ne sont pas indépendantes des dispositifs de sortie, il est nécessaire d'encapsuler les spécificités dans des procédures dites d'interfaces entre la partie générale du programme de communication et les organes de sortie. Un changement de matériel ne demande que la révision des procédures d'interfaçage.

Conclusion

Fonctionnant sur un micro-ordinateur du commerce et autour d'un noyau du langage Forth, le logiciel de communication peut être généralisé et structuré de la même manière pour la plupart des cas dans l'aide à la communication pour handicapés. La structuration se fait :

- pour la partie de données, en définissant la structure générale du pavé et du menu;
- pour la partie de programmes, en généralisant la communication par une procédure principale, et en définissant des modules changeables pour les fonctions de base de la communication.

La structure logicielle ainsi définie est à notre avis assez universelle et donc adaptative à la plupart des cas de figure dans le domaine de l'aide à la communication. Cela est vrai tant sur le plan de données que sur le plan de programmes.

La structure permet la génération automatique de logiciels de communication satisfaisant des besoins particuliers après une particularisation du contenu de la structure. Nous allons étudier dans le chapitre suivant les concepts généraux de la génération de logiciels de communication à partir de cette structure.

CHAPITRE III :

**GENERATION AUTOMATIQUE DE LOGICIELS DE
COMMUNICATION POUR HANDICAPES -- CONCEPTS
GENERAUX**

Introduction

Nous nous proposons d'étudier dans ce chapitre du mémoire la conception générale d'un système de génération automatique de logiciels de communication pour handicapés. Le système est basé sur les concepts et les structures matérielles et logicielles des communicateurs, que nous avons étudiés dans les deux chapitres précédents.

D'abord, nous résumons les approches suivies dans le domaine de la production automatique de logiciels d'application, cadre de ce mémoire.

Ensuite, nous détaillons ensuite une approche du domaine, appelée "progiciels individualisables", dont nous particularisons un système de génération avec un progiciel individualisable de communication pour handicapés.

Enfin, la conception générale du progiciel est étudiée en détail. Il s'agit de l'architecture générale du système, de son fonctionnement, et des différentes phases à suivre dans la production du logiciel de communication. L'accent est mis sur la spécification interactive des besoins via un questionnaire et un éditeur spécialisé, et sur la modularisation du logiciel de communication.

III.1 Production automatique de logiciels d'application [VAN. 82b] [KRA. 82]

La nécessité de la production automatique de logiciels d'application est mise en évidence par le fait que les logiciels mis en oeuvre de façon traditionnelle sont onéreux, mal documentés, peu modifiables et peu réutilisables. Un effort important a donc été mené ces dernières années pour développer des systèmes d'assistance à la production de logiciels, ou ateliers logiciels [DRO. 82] [LER. 82a] [LER. 82b]. Il s'agit des systèmes automatisant plus ou moins les activités menées tout au long du cycle de vie des logiciels. En parallèle avec ces systèmes, des méthodes ou approches se sont développées. On distingue souvent les approches suivantes :

- progiciels individualisables,
- langages de très haut niveau,
- programmation évolutive,
- programmation par l'exemple et
- systèmes à base de connaissances ou systèmes experts.

Certaines combinaisons sont possibles.

III.1.1 Progiciels individualisables [VAN. 82b]

Un progiciel individualisable est basé sur un ensemble de modules fortement paramétrés et couvrant les variantes des fonctions du domaine d'application. Une combinaison utile de ces modules donne un programme individualisé satisfaisant aux besoins d'un utilisateur particulier.

La spécification des besoins se fait souvent via un questionnaire arborescent. Suivant les réponses de l'utilisateur, le programme qu'il souhaite s'obtient alors par sélection de modules et paramètres correspondants.

Nous suivons cette approche pour la conception de notre système de génération. Elle sera étudiée en détail plus loin.

III.1.2 Langages de très haut niveau [VAN. 82b] [CHE. 84]

Les langages de très haut niveau possèdent souvent les caractéristiques suivantes :

- Mode d'expression non procédural

L'ordre d'apparition des instructions dans un programme rédigé en langage de très haut niveau n'a aucune importance. Le programmeur est libéré des contraintes de séquences dans la description de son application.

- Opérations sur des agrégats des données

Les traitements sont directement exprimés en termes d'opérateurs portant sur des collections de données. Le programmeur ne doit expliciter ni des itérations sur les éléments des collections, ni des détails d'organisation ou mode d'accès aux éléments.

- Références associatives

On accède à une donnée par simple description de certaines de ses propriétés, sans avoir à se préoccuper des chemins d'accès à cette donnée, ni de l'algorithme effectuant la recherche.

A l'aide d'un langage de très haut niveau dont la structure reflète la structure de l'application, et dont les constructions sont associées aux concepts fondamentaux du domaine, l'utilisateur, qui est souvent non programmeur, mais expert dans le domaine d'application, spécifie son application en termes de concepts du domaine. Il peut ainsi se concentrer sur la logique de son application.

Le générateur de programmes est dans ce cas un compilateur qui accepte comme entrée un programme source de très haut niveau, et qui sort le programme objet correspondant. Le compilateur doit effectuer des optimisations substantielles pour pouvoir produire du code objet suffisamment efficace.

Cette approche offre une grande flexibilité vis à vis des besoins spécifiques d'un utilisateur particulier, puisque cela dépend maintenant seulement du pouvoir d'expression du langage. Néanmoins, l'utilisateur est supposé capable de former son application dans le langage, et il doit prendre l'initiative.

III.1.3 Programmation évolutive [VAN. 83a]

L'approche "programmation évolutive" peut être considérée comme un compromis entre l'approche "progiciels individualisables" et l'approche "langages de très haut niveau".

Un système de production de logiciels est basé sur un ensemble des programmes de base (ou "plans") qui réalisent les fonctions standards ou types du domaine d'application, et sur un ensemble de règles de transformation de programmes.

Les programmes générés sont alors obtenus par transformation à partir de l'ensemble des programmes de base au moyen d'essais, modification et généralisation. L'utilisateur et le système coopèrent à ce processus de mise au point de très haut niveau dans un environnement interactif.

L'ensemble des programmes de base est souvent rédigé dans un langage de très haut niveau. Les modifications sur ces programmes suivant la spécification des besoins sont possibles, alors que ce n'est pas le cas pour les progiciels individualisables écrits dans un langage de niveau plus bas.

Il peut exister une sorte de programmes de niveau supérieur, i.e. un méta-programme, qui explicite la manière de composer certaines règles de transformation en vue d'obtenir un programme généré efficace à partir de l'ensemble des programmes de base. Le méta-programme décrit la façon de modifier les programmes de base et il est à son tour modifiable par l'utilisateur au moyen d'une interface de haut niveau.

III.1.4 Programmation par l'exemple [VAN. 83a] [BIE. 76]

L'approche "programmation par l'exemple" consiste à produire des programmes à partir de spécifications décrivant quelques exemples typiques des résultats que l'utilisateur souhaite obtenir, et les valeurs de données correspondantes à partir desquelles ces résultats doivent être produits.

Après la soumission de quelques exemples de relations d'entrée-sortie, le système, basé sur des informations à propos du domaine d'application et du langage cible, analyse les relations d'entrée-sortie, essaie de les généraliser pour en dégager une loi générale de correspondance, et dérive un programme adéquat à partir de cette loi. La généralisation de la loi de correspondance est parfois rendue possible et facilitée grâce à des interactions supplémentaires avec l'utilisateur (pour obtenir de nouveaux exemples). Des règles heuristiques et des informations concernant le domaine d'application (les structures de données et les propriétés de fonctions propres au domaine) sont souvent appliquées pour résoudre des ambiguïtés possibles au cours de la génération.

Une variante intéressante de cette approche consiste à spécifier au système en plus de ces paires d'entrée-sortie, des exemples de traces de traitement qui décrivent des suites de valeurs intermédiaires prises par les données en cours de traitement. Ces traces permettent d'alléger considérablement la tâche de synthèse.

La façon de spécifier un programme paraît naturelle et aisée à l'utilisateur.

III.1.5 Systèmes à base de connaissances [VAN. 83b] [HEW. 75] [LAU. 82]

Cette approche utilise des techniques relevant du domaine de l'Intelligence Artificielle. Il s'agit en effet d'un système expert de production de logiciels. Il fait usage intensif d'une base de connaissances, dans laquelle on distingue :

- des connaissances sur la programmation,
- des connaissances sur le domaine d'application,
- des connaissances sur la mise au point et
- des connaissances sur le comportement de l'utilisateur et sur l'interface utilisateur-système.

Ces connaissances sont souvent organisées de façon modulaire à différents niveaux d'abstraction avec différentes représentations suivant les niveaux.

Un programme appelé moteur d'inférence se charge du raisonnement. Il applique des connaissances au problème particulier de l'utilisateur pour sa résolution. Les spécifications de l'application se trouvent dans une base de données, la base de faits.

A l'aide d'un système à base de connaissances, la production d'un programme suit dans la plupart des cas les phases suivantes de traitement :

- Acquisition du problème

L'utilisateur spécifie son application via une interface fortement interactive, par exemple dans une langue naturelle réduite et sa spécification est alors convertie en un modèle particulier.

- Transformation en procédures

Le modèle particulier est traité à l'aide des connaissances propres au domaine d'application et transformé en modèle de solution de haut niveau.

- Vérification du modèle

Le système interagit à nouveau avec l'utilisateur pour vérifier la conformité du modèle à ses besoins, et pour effectuer d'éventuelles modifications sur le modèle.

- Programmation automatique

L'application de règles de dérivation et d'optimisation au modèle permet de produire un programme pouvant satisfaire les besoins de l'utilisateur.

La structure du système (l'indépendance des fragments de connaissances) facilite l'extension et la modification du système. Une interface permet à un expert du domaine d'application de modifier, de supprimer et d'ajouter des connaissances. Le système peut ainsi suivre facilement l'évolution du domaine. La structure offre également une grande flexibilité à l'égard des besoins particuliers, grâce à l'organisation hiérarchique et modulaire de la base de connaissances.

III.2 Etude détaillée de l'approche "progiciels individualisables" [VAN. 82b]

Nous nous proposons dans ce paragraphe d'étudier en détail la production de logiciels dans l'approche "progiciels individualisables", car c'est ce que nous avons adopté pour notre système de génération de logiciels de communication.

III.2.1 Principe de génération

L'idée de l'approche est de représenter par des modules pré-programmés les variantes des fonctions de base du domaine d'application, et de produire le programme d'un utilisateur particulier par combinaison de modules répondant aux besoins.

Un progiciel individualisable est basé sur un ensemble de modules de programmes fortement paramétrés et les valeurs des paramètres correspondants. Cet ensemble comprend les codes alternatifs pour toutes les variantes prévisibles des fonctions de base du domaine d'application; il est donc censé couvrir la totalité du domaine. Une combinaison pertinente de modules et de valeurs de paramètres constitue un programme particulier correct dans le domaine. Celui-ci peut alors être "généralisé" par sélection et assemblage de modules et de valeurs de paramètres correspondants.

L'ensemble de modules est souvent rédigé dans le langage cible du domaine d'application. Il peut être écrit dans un langage de haut niveau, ce qui augmente, au risque de perte en efficacité, la portabilité du système, et facilite la maintenance et l'extension.

L'approche "progiciels individualisables" utilise souvent un questionnaire préétabli pour l'acquisition des besoins du programme à générer. Il y a dans le questionnaire un menu standard à choix multiple pour chaque fonction couverte par le domaine d'application. Les questions portent généralement :

- sur les flux logiques possibles de l'application,
- sur sa configuration matérielle et logicielle,
- sur les formats de représentation d'informations pour l'environnement de l'application,
- et sur la quantification des données à traiter.

Le questionnaire doit être rédigé dans un langage compris par l'utilisateur, souvent en termes de concepts du domaine d'application. Les détails informatiques d'implémentation ne doivent pas figurer dans le questionnaire. Des explications peuvent être associées aux questions pour les utilisateurs non expérimentés.

III.2.2 Phases de génération d'un produit logiciel

Le processus de génération d'un produit logiciel comprend une phase de spécification des besoins, une phase de production du programme et une phase éventuelle de production de la documentation du produit.

1) Spécification des besoins de l'application

La spécification des besoins du programme à produire se fait grâce au questionnaire préétabli. L'utilisateur reçoit le questionnaire, et répond suivant le cahier des charges de l'application, aux questions posées, éventuellement dans un environnement interactif.

Le système analyse les réponses de l'utilisateur, détecte les incohérences et les incompatibilités entre les réponses, et n'accepte que les réponses jugées "correctes". La hiérarchie et l'ordre des questions sont prédéterminés. Cependant, il est possible que le système sélectionne des questions à poser en fonction de réponses déjà retenues.

Au fur et à mesure que la spécification se déroule, le système construit à partir des réponses le modèle particulier du logiciel à produire, sous une forme interne. Ce modèle détermine d'une part le sous-ensemble de modules à sélectionner parmi l'ensemble de modules préprogrammés et les valeurs de paramètres, d'autre part l'ensemble de données propres à l'application. Il est utilisé pour la production du programme et la production de la documentation.

2) Production du programme

Suivant le modèle particulier, le système sélectionne le sous-ensemble de modules, substitue les valeurs des paramètres de ces modules, génère les structures de données de l'application, et assemble ces modules et ces données pour obtenir un programme dont les comportements peuvent satisfaire les besoins spécifiés.

Si le programme généré n'est pas rédigé dans le langage cible de l'application, une transformation, par exemple une compilation croisée, est nécessaire.

Les opérations dans cette phase sont automatisées dans la plupart des systèmes, l'utilisateur n'a pas à y intervenir.

3) Production de la documentation

Le système peut à la demande de l'utilisateur produire la documentation du produit généré, grâce au modèle particulier du logiciel. La documentation peut se présenter sous différentes formes : mode d'emploi du logiciel pour l'utilisateur, rapport informatique pour l'utilisateur du système de génération.

Le modèle particulier décrivant entièrement le logiciel généré peut être archivé pour d'éventuelles réutilisations ultérieures : modification, adaptation et extension.

III.2.3 Mérites et limites de l'approche

L'approche "progiciels individualisables" est la seule à être opérationnelle et commercialisée, puisqu'elle est aussi plus proche des pratiques traditionnelles dans la programmation.

Elle permet de produire automatiquement des logiciels d'application "taillés sur mesure" dans un domaine d'application bien circonscrit, avec des gains considérables en temps et en efforts.

La méthodologie de spécification via un questionnaire est d'un grand intérêt. Elle ne demande aucun formalisme et l'initiative est toujours prise par le système, non par l'utilisateur; celui-ci n'a qu'à répondre aux questions en choisissant la ou les options appropriées parmi celles présentées. Cette facilité est jugée d'autant plus nécessaire que l'utilisateur est non expérimenté, ou occasionnel.

Le défaut majeur d'un progiciel individualisable est le manque de flexibilité vis à vis des besoins spécifiques d'un utilisateur particulier. Il peut arriver qu'un cas particulier d'exigence ne soit pas couvert par le progiciel. Ce défaut est inhérent à l'approche, du fait que les différents types d'informations, qui concernent l'application et la programmation, sont tous "figés" en une seule représentation, souvent de bas niveau : les modules paramétrés.

Un autre défaut de l'approche est l'inefficacité de certains programmes générés : des redondances de données, de programmes et d'opérations sont parfois inévitables.

III.3 Justification du choix de l'approche

Le système de génération de logiciels de communication, qui fait l'objet de cette étude, s'appuie sur l'approche "progiciels individualisables". Les raisons pour lesquelles nous la préférons autres paraissant pourtant plus performantes peuvent être les suivantes :

- Faisabilité

L'approche est la plus proche des pratiques traditionnelles en génie logiciel, et faisable pratiquement. Elle se trouve la seule à être opérationnelle et commercialisée.

- Facilité d'utilisation du système de génération

L'approche offre une grande facilité d'utilisation du système de génération. Le système est destiné à être manipulé par des personnels des hôpitaux ou des centres de rééducation, qui peuvent ne connaître ni l'informatique, ni tous les concepts du domaine de l'aide à la communication. Ils peuvent apprécier beaucoup de spécifier un communicateur via un questionnaire avec des options de réponses clairement définies.

- Faible dimension du système de génération

L'approche est la plus simple parmi toutes celles étudiées. Le système de génération peut être d'une dimension faible, donc implémentable sur

micro-ordinateur, matériel courant dans les hôpitaux et les établissements de rééducation.

- Facteurs du domaine d'application

Les caractéristiques du domaine d'application jouent un rôle important dans la prise de décision du choix de l'approche. L'aide à la communication pour handicapés présente un domaine bien délimité dans lequel les opérations sont relativement simples et les variantes des fonctions de base sont prévisibles avant la mise en oeuvre du système de génération. De plus, le logiciel de communication peut être structuré de façon identique pour tous les systèmes de communication, tant sur le plan des programmes exécutables que sur le plan des structures de données (pavé et menus). Cette structuration généralisée facilite beaucoup la tâche de génération. Le logiciel d'aide à la communication se prête bien à être généré par l'approche "progiciels individualisables".

Comme nous l'avons analysé, les défauts majeurs de l'approche sont le manque de flexibilité et l'inefficacité éventuelle de programmes générés (voir III.2.3). Cependant, dans le cas d'aide à la communication pour handicapés, les effets de ces défauts restent faibles et donc tolérables :

- Manque de flexibilité :

Les données représentent la plus grande partie d'un logiciel d'aide à la communication. Au moment de la génération, la part la plus importante d'effort est consacrée à la génération de ces données, qui consiste simplement à remplir une structure prédéfinie avec les valeurs spécifiques de l'application. Un éditeur de données peut accomplir cette tâche. L'éditeur de pavé couvre facilement tous les types de pavés.

De plus, dans le cas où le système de génération s'avère incapable de générer une fonction très particulière, la modification du logiciel généré ou l'ajout de nouveaux modules au système de génération paraissent faisables, étant donné la simplicité de la partie "programmes" du logiciel et le fait que ces programmes sont codés en langage Forth, qui se prête bien à l'écriture de programmes modulaires modifiables.

- Inefficacité de certains programmes générés

Le logiciel généré étant de faible dimension et la rapidité d'exécution du logiciel n'étant pas importante dans l'application, l'effet de l'inefficacité reste mineur.

III.4 Progiciel individualisable d'aide à la communication pour handicapés

Nous nous proposons dans ce paragraphe d'étudier l'architecture et le fonctionnement général du progiciel d'aide à la communication pour handicapés. Nous examinons les différents composants logiciels, leur rôle dans le système, et les différentes phases dans le développement du logiciel de communication.

III.4.1 Schéma d'architecture et de fonctionnement [HE 84]

La figure 3.1 montre le schéma d'architecture et de fonctionnement du système de génération de logiciels d'aide à la communication.

Le progiciel fonctionne suivant le principe général de l'approche, avec quelques spécificités liées au domaine d'application :

- la spécification des besoins de l'application se fait via un questionnaire et un éditeur de données (pavés menus et dictionnaires);
- le logiciel source est produit, à partir des spécifications, par sélection et paramétrage de modules prédéfinis;
- le logiciel peut être testé sur le système de génération grâce à un simulateur de la machine cible;
- le logiciel source est traduit en code objet à l'aide d'un traducteur;
- la documentation du logiciel est rédigé par un générateur de documents;
- les informations issues de la spécification, qui constituent le modèle du logiciel, peuvent être archivées pour des modifications et extensions ultérieures.

Ces points sont décrits plus loin.

Examinons maintenant les fonctions des différents composants de l'architecture du système :

- Questionnaire

Ce composant est utilisé pour la spécification des besoins : configuration matérielle et logicielle, logique de l'application, etc; et il consiste en un ensemble de questions organisées hiérarchiquement.

- Editeur de données

Ce composant est un programme spécialisé pour édition interactive de pavés et menus et de dictionnaire. Il est utilisé pendant la phase de spécification.

- Modules paramétrés

Il s'agit de l'ensemble des modules de programmes exécutables qui couvrent l'ensemble des fonctions de base et leurs variantes dans l'aide à la communication; ils constituent le logiciel à générer.

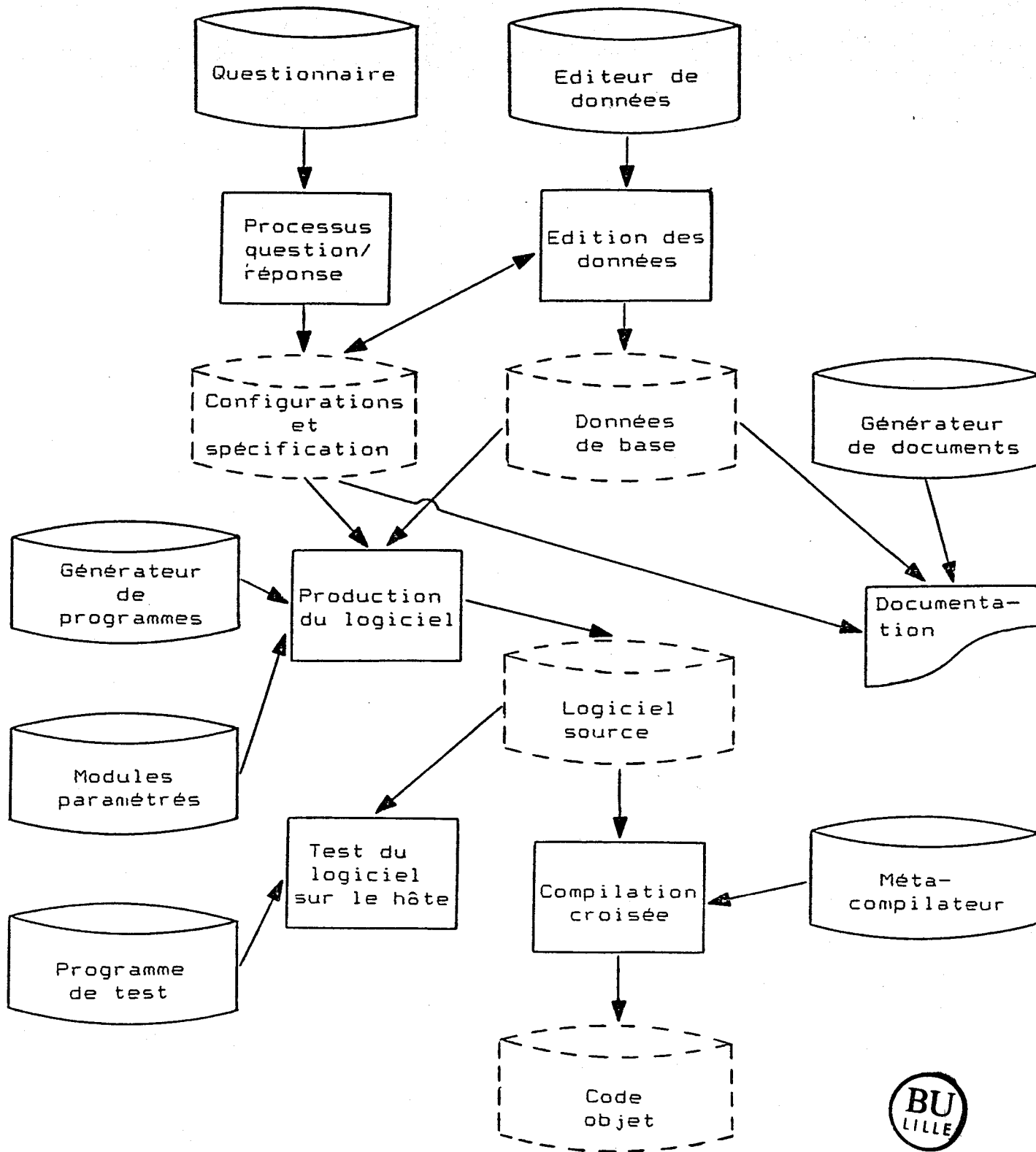


Fig. 3.1 Architecture et fonctionnement du système de génération

- Générateur de programme

Ce composant se charge de sélectionner et d'assembler les modules constitutifs et les données de base, en fonction des spécifications.

- Programme de test

Ce programme est utilisé pour compiler et tester le logiciel généré sur le système hôte, il peut être un simulateur du système cible.

- Traducteur

Ce programme produit le logiciel généré sous forme implémentable sur le système cible; et il peut être un compilateur croisé.

- Générateur de documents

Ce programme produit le document du logiciel généré.

III.4.2 Spécification des besoins

L'utilisateur spécifie, d'après un cahier des charges, les besoins de son application à l'aide d'un questionnaire prédéfini et d'un éditeur interactif de données.

Les spécifications saisies constituent le modèle particulier du logiciel de l'application.

III.4.2.1 Spécification via le questionnaire [HE 84]

Le questionnaire constitué de menus et sous-menus hiérarchisés est utilisé pour la saisie des informations sur les configurations du système cible : organe d'accès, organes de sortie, écran vidéo et système logiciel, et sur le choix de certaines fonctions de base.

On peut y trouver des interrogations :

- sur l'organe d'accès :

L'organe d'accès détermine les méthodes de sélection possibles dans le communicateur; ce peut être un interrupteur simple, un clavier ou un équivalent. Dans le cas d'un clavier, il faut spécifier le nombre de touches ou le nombre d'entrées distinctes.

- sur les organes de sortie :

Le type des organes de sortie est multiple. Les organes de sortie typiques sont l'imprimante, l'écran vidéo, le synthétiseur de parole et les appareils de contrôle de l'environnement. Pour chaque type d'organe de sortie, il peut y avoir des questions spécifiques. Les informations sur les organes de sortie permettent de déterminer le type des messages finaux possibles dans le communicateur.

- sur l'écran de visualisation :

- la dimension ou résolution de l'écran,
- le type de l'écran (texte, graphique, texte et graphique, couleur, monochrome, etc),
- la configuration de l'écran : la fenêtre de menu, celle ou celles de message et celle de renseignement (leur dimension, leur position et leur couleur de fond).

La définition de l'écran se effectue de manière interactive.

- sur l'ordinateur cible :

- la quantité de mémoire vive disponible,
- l'existence d'un Forth.

- sur les méthodes d'accélération :

Il est possible de proposer à l'utilisateur des méthodes d'accélération pour chaque type de message primaire. Les réponses de l'utilisateur servent à la sélection des modules et des structures d'accélération correspondantes.

- sur les types de messages primaires possibles

La façon de visualiser et de manipuler un message primaire dépend du type du message. Les types que l'on rencontre souvent sont : textes rédigés dans une langue usuelle par exemple le Français simplifié, symboles BLISS, et dessins composés de pavés graphiques, etc.

Les informations de configuration du système cible sont affectées aux paramètres correspondants dans le module de paramètres "configurations et spécifications", et utilisées pour l'édition de données et la sélection de modules paramétrés.

III.4.2.2 Edition de données de base de la communication

Les données de base de la communication constituent la plus importante partie d'un logiciel de communication. La spécification des besoins d'une application comprend ainsi l'édition des données de base. Celle-ci sont les constituants primaires (pavés) et les menus et sous-menus, et un éventuel dictionnaire.

Pour chaque constituant primaire, il faut spécifier :

- sa dimension,
- son contenu,
- et sa fonction.

Pour chaque menu, les informations à spécifier sont :

- les constituants primaires qu'il utilise,
- la position de chacun des constituant dans la fenêtre de menu,
- la méthode sélection et le mode de désignation associés au menu.

Pour le dictionnaire dont la structure est à définir, il est préférable que nous puissions le construire à partir d'un dictionnaire de base standard prédéfini. Un environnement interactif permet à l'utilisateur de sélectionner des mots du dictionnaire standard, et de composer les mots qui ne figurent pas dans ce dernier.

Les informations acquises dans l'édition sont enregistrées dans le module "données de base". De plus, l'éditeur fournit au module "configurations et spécifications" les informations concernant les méthodes de sélection, les pavés de commande, les types de pavés, etc.

L'éditeur utilise des informations du module "configurations et spécifications" pour le format de la fenêtre de menu et le choix de questions conditionnées au moment de l'édition.

III.4.2.3 Modèle utilisateur

A l'issue de la phase de spécification, les informations acquises se trouvent enregistrées dans les deux modules "configurations et spécifications" et "données de base". Ceux-ci constituant le modèle utilisateur particularisé caractérisent entièrement le logiciel répondant aux spécifications de l'utilisateur. Le module "configurations et spécifications" contient les informations propres à la configuration du système cible, et les informations concernant la sélection et le paramétrage de modules parmi l'ensemble de modules prédéfinis. Le module "données de base" décrit l'ensemble des données structurées propres à l'application particulière (pavés, menu et dictionnaire).

Ce modèle particularisé est par la suite utilisé pour la production du logiciel qu'il décrit, et pour la production des documents du logiciel. Il peut être archivé dans une banque "communicateurs" pour d'éventuelles réutilisations : modification, extension, et adaptation à un autre communicateur.

III.4.3 Production du logiciel source

Le composant "générateur de programme" produit automatiquement un logiciel de communication pouvant répondre aux spécifications, à partir du modèle utilisateur particularisé issu de la phase de spécification, et de l'ensemble de modules prédéfinis paramétrés.

Le processus de génération du logiciel consiste en deux étapes :

- la génération de la partie "programmes" à partir du composant "configurations et spécifications" et de l'ensemble de modules prédéfinis;
- et la génération de la partie "données" à partir du composant "données de base".

Le logiciel généré est mémorisé dans un fichier sous forme de code source du Forth. L'utilisateur peut soit le tester sur le système hôte, soit lui faire subir une transformation, par exemple une compilation croisée, pour pouvoir le charger par la suite dans le système cible.

III.4.3.1 Ensemble de modules prédéfinis

La base du progiciel individualisable d'aide à la communication est un ensemble de modules de programme prédéfinis et fortement paramétrés, dont les combinaisons pertinentes doivent couvrir toutes les possibilités des variantes du domaine d'application.

1) Critères de décomposition [PAR. 72] [PAR. 85]

La décomposition de l'application en modules et la conception de ces modules sont des tâches difficiles et complexes. La fiabilité, la modifiabilité et l'extensibilité du système dépendent beaucoup de la qualité de la décomposition et de la conception.

Comme pour tous les systèmes informatiques, les critères suivants de décomposition s'appliquent au progiciel individualisable :

- un programme doit être fortement hiérarchisé, décomposé en modules; chaque niveau de la hiérarchie réalise un degré d'abstraction;
- chaque module doit cacher ses détails d'implémentation aux autres modules, tout en présentant une interface fonctionnel bien défini, de façon à ce que le changement de l'implémentation du module ne demande aucune révision des autres modules;
- il doit y avoir un faible degré de couplage entre modules de manière à ce que les différentes fonctions réalisées soient aussi indépendantes que possible, et par conséquent facilement insérables ou retirables.

De plus, dans le cas du progiciel individualisable, il doit y avoir des correspondances bien définies et localisées entre les options ou variantes des fonctions de base du domaine d'application et les modules qui les réalisent. Chaque option ou variante doit être indépendante des autres. Il est très important de bien distinguer les modules communs à tout application des modules d'option ou de variante. Le changement d'un module optionnel n'affecte jamais les modules communs.

2) Modules des fonctions d'option ou de variante.

Dans le cas de l'aide à la communication, nous pouvons identifier de façon non exhaustive les modules fonctionnels d'option ou de variante pour les fonctions suivantes :

- Sélection de constituants primaires

Les modules possibles de sélection sont :

- sélection sans codage,
- sélection avec codage ligne-colonne (ou colonne-ligne),
- sélection avec codage dichotomique mixte,
- sélection par balayage en série,
- sélection par balayage ligne-colonne (ou colonne-ligne),
- et sélection par balayage dichotomique mixte.

- Commandes de contrôle d'édition des messages

Les commandes de contrôle d'édition des messages dépendent de la nature du message que l'utilisateur souhaite construire, et des besoins de ce dernier. Elles varient de simples déplacement du curseur jusqu'à toutes les fonctions d'édition offertes par un logiciel de traitement de texte.

- Commandes de contrôle de l'environnement

Les commandes de contrôle de l'environnement dépendent de la configuration de l'environnement de l'utilisateur handicapé. On configure un communicateur selon l'environnement de la communication.

- Affichage des messages primaires

La manière de visualiser un message primaire dépend du type du message. Le module correspondant est choisi en fonction du type. Les types de message que l'on rencontre souvent sont : texte alphanumérique, suite de symboles BLISS et dessin composé de pavés graphiques.

- Accélération de la construction des messages primaires

Il s'agit des méthodes d'accélération dynamiques. Il est possible d'avoir des modules d'accélération à choix en fonction du type de message, et de certains autres facteurs : le dictionnaire, l'espace mémoire disponible.

- Transformation de messages

Pour chaque couple "message primaire-message final", il doit y avoir un module de transformation. Le système doit choisir le module de transformation en fonction du doublet "message primaire-message final" de l'application.

Pour chaque module d'option, il existe dans le composant "configurations et spécifications" une ou des variables booléennes qui conditionnent l'inclusion du module dans le logiciel à générer.

III.4.3.2 Génération de la partie "programmes"

Les programmes proprement dits du logiciel de communication sont automatiquement générés par le composant "générateur de programmes" à partir de l'ensemble des modules prédéfinis et des informations contenues dans le composant "configurations et spécifications". Ils sont rangés dans un fichier sous forme de code source du Forth dans l'ordre de leur compilation.

Pour générer ces programmes, le générateur :

- sélectionne parmi les modules d'option ou de variante ceux dont la condition de sélection a été vérifiée après l'évaluation de la condition,

- substitue les paramètres des modules sélectionnés avec les valeurs correspondantes issues de la spécification, comme par exemple les paramètres du format de l'écran du système cible,
- et assemble ces modules et les modules communs.

III.4.3.3 Génération de la partie "données"

Les données de l'application, i.e. les pavés, les menus et/ou le dictionnaire, sont générées automatiquement par le générateur de programmes à partir des informations contenues dans le composant "données de base", et ajoutées aux programmes générés.

Il s'agit de convertir ces informations de représentation interne en une forme implémentable par la suite, par exemple sous forme de codes ASCII. Des références à des procédures d'aide à l'implémentation des données peuvent être insérées, et elles permettent l'implémentation de ces données dans le système cible sous la structure définie.

III.4.4 Test du logiciel dans le système hôte

Avant de charger le logiciel généré dans le système cible, nous pouvons le tester sur le système de génération à l'aide d'un programme de test. Pour ce faire, nous procédons de la manière suivante :

- L'écran du système hôte est reconfiguré pour simuler l'écran du système cible;
- L'organe d'accès est simulé par le clavier du système hôte;
- Une imprimante ou un synthétiseur de voix sert à simuler un organe de sortie pour les messages alphanumériques;
- L'environnement logiciel du système cible peut être simulé :
 - soit par un simulateur du système, qui offre un environnement logiciel identique au système cible, c'est le meilleur cas;
 - soit par le système Forth de l'hôte, du fait que le noyau du logiciel est exécutable dans le système hôte après une adaptation des interfaces;
- Le programme de test se charge alors de compiler le logiciel et l'exécuter; des facilités de contrôle de l'exécution et de dépannage peuvent être développées et utilisées par l'utilisateur expérimenté.

Le test sur le système hôte permet, malgré ses limites, de valider dans certaine mesure le logiciel généré : des non conformités aux besoins, des erreurs de spécification et des mauvaises configurations peuvent être identifiées. En cas d'erreur, l'utilisateur peut retourner à la phase de spécification, et rappeler le questionnaire ou l'éditeur de données pour corriger les erreurs, ce qui permet de gagner un gain en temps de développement.

III.4.5 Production du logiciel objet

Le logiciel source généré est rédigé dans le langage Forth du système de génération. Il doit être transformé, par un programme de traduction, ou "traducteur", en code objet pour le système cible. Nous entendons par code objet le code implémentable sur le système cible, qui peut être du Forth compilé ou source de ce système.

S'il y a dans le système hôte un compilateur croisé pour le système cible, la transformation devient une compilation croisée. Le code objet résultant peut être chargé soit dans une EPROM soit sur une disquette.

En cas de non existence d'un compilateur croisé, nous pouvons procéder à une transformation source-source (Forth de l'hôte-Forth de la cible), étant donné la faible différence entre les mots de base de différents langages Forth. Le code résultant est du code source du Forth du système cible, et donc compilable dans ce système.

Une autre possibilité est de mettre dans une même cartouche d'EPROM ou PROM le logiciel de communication compilé et le noyau d'un Forth, i.e. l'interpréteur interne de ce Forth, pour une machine cible où il n'y a pas de système Forth.

III.4.6 Documentation du logiciel de communication

En dehors du logiciel de communication, le système de génération peut produire automatiquement des documents sur le logiciel généré, grâce à un générateur intégré de documents.

Le générateur de documents fonctionne suivant le même principe que celui suivi par le générateur de programmes : il existe un ensemble de documents correspondant à l'ensemble des modules prédéfinis; chaque module, commun ou optionnel, peut être accompagné d'une documentation standard et paramétrée; les informations contenues dans le modèle utilisateur individualisé sont utilisées pour la sélection et l'assemblage de morceaux de documents.

Pour l'utilisateur du logiciel de communication, la documentation comprend le mode d'emploi, la liste des pavés, la liste des menus, le graphe de relations entre les menus, etc.

Pour l'utilisateur désirant connaître et modifier le progiciel, la documentation consiste en l'organigramme de chacun des modules de programme, la description informatique des données de base, etc.

III.4.7 Archivage et réutilisation des spécifications

Le système de génération archive le modèle utilisateur individualisé de façon à ce que les informations contenues dans le modèle puissent être utilisées ultérieurement non seulement pour la maintenance et l'extension du logiciel, mais aussi pour d'autres applications.

La modification et extension du logiciel après une période d'utilisation consistent à reprendre simplement le modèle et à générer de nouveau un logiciel. Ceci permettra de suivre facilement l'évolution des besoins de l'utilisateur handicapé.

De plus, nous pouvons construire une vaste banque d'informations accessibles à tout utilisateur, dans laquelle il peut y avoir une base de "communicateur", une base de "pavés", une base de "menus", un dictionnaire standard, etc. Des utilitaires permettent divers accès aux informations. Ainsi, l'utilisateur peut spécifier son application en utilisant des informations existantes dans la banque, ce qui permet une mise en oeuvre plus rapide et facile des logiciels de communication.

Conclusion

Une brève étude a été faite sur les approches développées dans le domaine de la production automatique des logiciels d'application. L'approche dite "progiciels individualisables" a été retenue et appliquée à la conception du système de génération de logiciels d'aide à la communication pour handicapés. Elle convient bien, malgré ses défauts inhérents, au domaine traité.

Le progiciel individualisable d'aide à la communication pour handicapés permet la mise en oeuvre aisée et rapide, par un non programmeur, des logiciels d'aide à la communication "taillés sur mesure", et donc adaptés aux besoins spécifiques des handicapés.

Quelques expériences ou leçons que nous pouvons tirer de la conception de ce progiciel méritent d'être remarquées :

- Il y a nécessité une expertise grande et complète dans le domaine d'application, sinon, le système risque d'être mal adapté aux souhaits des utilisateurs.
- Une méthodologie de conception systématique du questionnaire de spécification paraît fort utile; elle aide à décomposer le domaine d'application, à identifier les concepts communs et les variantes, et à organiser le questionnaire afin qu'il soit complet, souple et naturel.
- La décomposition de l'application en modules, et la conception de l'ensemble des modules dont les combinaisons pertinentes doivent produire un ensemble cohérent et suffisamment complet de solutions pour le domaine d'application, représentent des tâches très difficiles et complexes; il nous manque des méthodologies systématiques et rigoureuses pour l'accomplissement de ces tâches.

Une réalisation concrète de ce travail est présentée par la suite.

CHAPITRE IV :

IMPLEMENTATION DU SYSTEME DE GENERATION DE COMMUNICATEURS POUR HANDICAPES

Introduction

Après avoir étudié les concepts généraux d'un système de génération automatique de communicateurs pour handicapés, nous nous proposons de présenter dans ce chapitre les différentes implémentations effectuées au cours de la poursuite du projet.

Deux versions du système ont été implémentées dans différentes configurations : l'une fonctionne sur un micro-ordinateur APPLE II et génère des communicateurs pour une machine portable (HHC du Panasonic); l'autre qui est la version compatible IBM-PC produit des communicateurs pour compatible IBM-PC.

Pour commencer, nous décrivons les différentes configurations matérielles et logicielles, puis nous résumons la démarche de la génération d'un logiciel de communication : spécification et génération.

Des études détaillées sont ensuite faites sur l'édition de communicateur (spécification des besoins) et sur la génération proprement dite.

Nous finissons ce chapitre en illustrant le fonctionnement du système à l'aide de quelques exemples de scénarios d'utilisation du système.

IV.1 Configurations matérielles et logicielles

Avant de décrire les différentes configurations, il est important de distinguer le système de développement, ou système hôte, du système cible. Le système hôte est le système dans lequel le programme de génération fonctionne. Le système cible est celui auquel le logiciel généré est destiné.

Pour la version APPLE II du programme de génération, le système hôte est un micro-ordinateur APPLE II, le système cible un micro-ordinateur portable (HHC de Panasonic). Pour la version IBM-PC, le système hôte et le système cible sont les mêmes types, un IBM-PC ou compatible.

IV.1.1 Configuration APPLE II-HHC

La première version du programme de génération fonctionne dans un environnement APPLE II-HHC. Un micro-ordinateur APPLE IIe est utilisé comme machine hôte, et un micro-ordinateur portable HHC de PANASONIC comme machine cible. La communication entre les deux machines est assurée par une liaison série RS-232.

Le logiciel de développement (voir le manuel "Host Operating System 4.1A"), établi en langage Forth, comprend les éléments suivants :

- un noyau du langage Forth, qui permet à la fois de programmer et de lancer des commandes pour activer les autres facilités du système;

- un éditeur de texte plein-écran avec des fonctions d'édition élaborées, qui peut servir à éditer des programmes pour la machine hôte et pour la machine cible;
- des facilités de gestion de fichiers;
- un émulateur de la machine cible HHC, qui permet d'exécuter du code compilé du SnapForth, le langage cible, sur la machine hôte;
- un compilateur croisé pour le langage SnapForth, qui sert à compiler des programmes écrits en SnapForth, et à créer du code ROMable;
- un chargeur de code compilé du SnapForth sur un programmeur d'EPROM.

La machine cible HHC est portable, tous ses modules sont rangés dans une mallette. Ils comprennent : un module central (processeur, clavier, 8 K octets de mémoire vive, afficheur à cristaux liquides), un adaptateur d'E/S, des interfaces RS-232 et MODEM. De plus, il existe de nombreux emplacements pour des cartouches de programmes.

Dans l'environnement cible, le langage de programmation est le SnapForth (voir le manuel du langage), dont la structure permet la ROMabilité de programmes, et l'exécution d'un programme à partir d'une cartouche insérée.

Une cartouche d'interpréteur SnapForth nous permet de compiler, sur la machine cible, des programmes sources développés sur la machine hôte et envoyés via une liaison série RS-232.

Un synthétiseur vocal lié à la machine cible via une liaison série RS-232 sert à l'organe de sortie pour tester des communicateurs générés. Le clavier de la machine cible simule l'organe d'accès du communicateur.

IV.1.2 Configuration IBM-PC

Écrit en PC/Forth+, cette version du programme fonctionne sur un IBM-PC ou un compatible. Elle génère des communicateurs pour la même machine.

Le langage PC/Forth+ (voir le manuel du langage), utilisé à la fois pour la génération et pour l'exécution, tourne sous le système d'exploitation MSDOS, et il fournit un environnement interactif de développement. Il comprend :

- le noyau du PC/Forth+,
- un éditeur plein-écran de blocs de programme,
- des facilités de gestion de fichiers hiérarchisés,
- des primitives graphiques couleurs (résolutions : 320x200 640x200 640x350 avec 2, 4 ou 16 couleurs),

- et plus de 300 K octets de mémoire vive accessibles par le dictionnaire du langage.

Une facilité intéressante offerte par le langage est la possibilité de créer du code compilé sous une forme relogeable (forme OVERLAY). Le logiciel de communication généré est sauvegardé sur disquette sous cette forme. Il peut s'exécuter automatiquement au démarrage de la machine.

Comme pour la version APPLE II, un synthétiseur vocal est utilisé comme organe de sortie, le clavier de la machine hôte comme organe d'accès.

IV.2 Architecture et principe de fonctionnement du système

La figure 4.1 présente l'architecture et le fonctionnement du système.

IV.2.1 Architecture du système

Les trois principaux éléments de base du système sont l'éditeur de pavés et menus, le générateur de communicateur et l'ensemble des modules pré-programmés.

L'éditeur de pavés et menus est un programme interactif. Il permet à l'opérateur de spécifier les besoins de son communicateur en définissant les données utilisées dans la communication : les menus et les pavés du communicateur.

L'ensemble de modules prédéfinis représentent des morceaux de programmes rédigés dans le langage cible (le SnapForth pour la version APPLLE II et le PC/Forth+ pour la version IBM-PC) et destinés à constituer les communicateurs. Il comprend les modules obligatoires pour tout communicateur, et les modules d'option. Ces derniers concernent les méthodes de sélection, les commandes de contrôle de l'environnement et les commandes de contrôle d'édition des messages.

Le générateur de communicateur fonctionne sur le système hôte et il sert à générer un logiciel de communication en fonction des spécifications saisies par l'éditeur, et à partir de l'ensemble de modules prédéfinis.

Le composant intermédiaire "pavés et menus" représente les spécifications du communicateur à générer.

Le composant "communicateur généré" est le logiciel de communication pouvant répondre aux besoins spécifiés. Dans la version APPLE II, il se trouve sous forme de code compilé ROMable, ou sous forme de code source du SnapForth. Dans la version IBM-PC, il est sauvegardé sous forme compilée relogeable (forme OVERLAY).

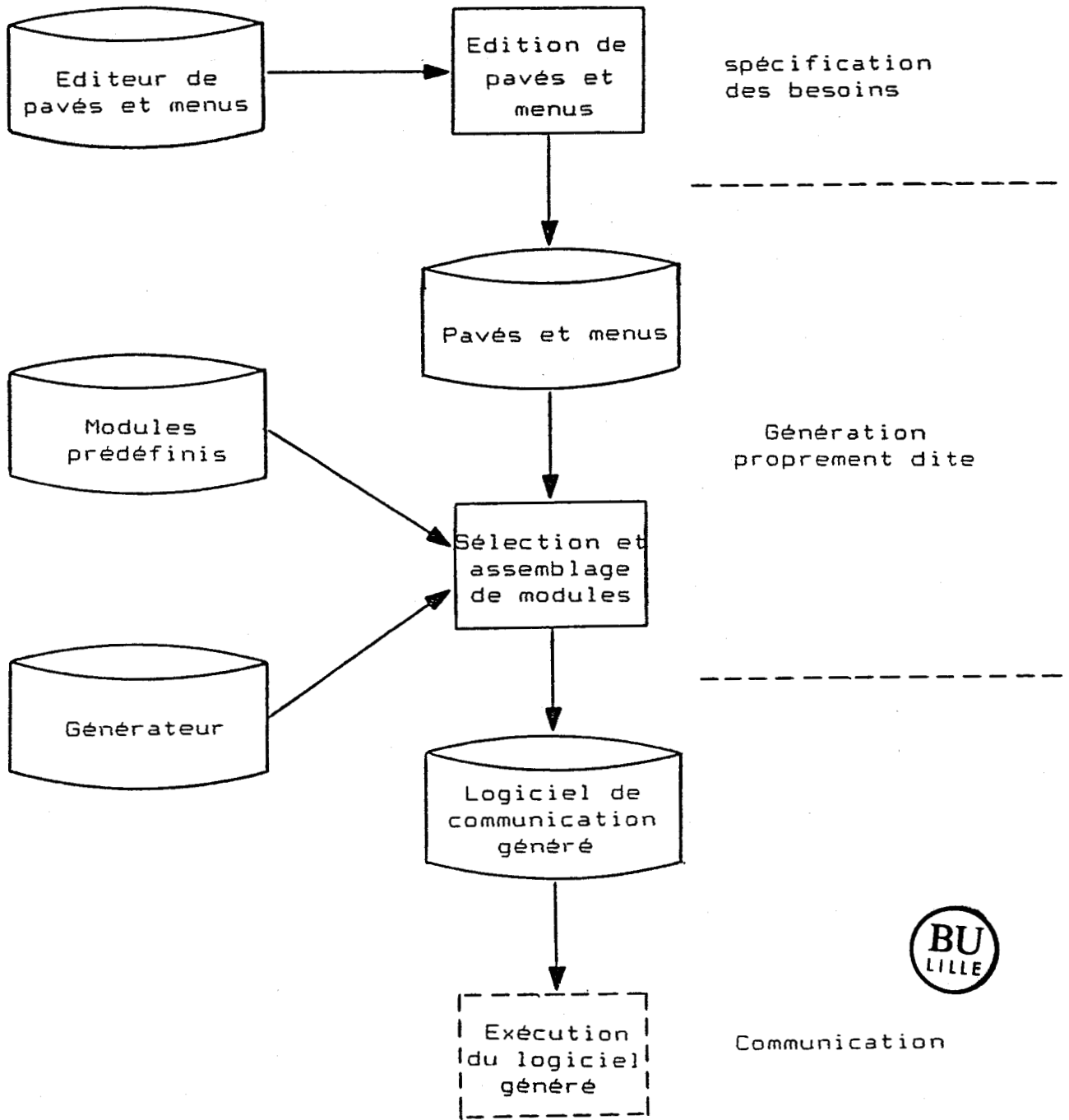


Fig. 4.1 Architecture et principe du fonctionnement du système de génération

IV.2.2 Processus de génération

Le système fonctionne suivant le cycle édition-génération-exécution.

Au premier niveau, le niveau d'édition ou de spécification, l'opérateur peut spécifier son communicateur à l'aide de l'éditeur de pavés et menus, qui lui fournit un environnement d'édition interactif. L'édition est une suite de constructions de pavé et de menu, elle est constamment guidée par la syntaxe des objets manipulés (pavés ou menus), les spécifications incohérentes sont rejetées par l'éditeur. L'opérateur n'a qu'à activer des commandes ou à répondre à des questions. Les spécifications acquises sont mémorisées dans des fichiers utilisés par la suite.

Au deuxième niveau, le générateur de communicateur est déclenché. Il se charge de construire un logiciel de communication en fonction des spécifications saisies précédemment. Ce logiciel est constitué des deux parties : les données et les programmes. La partie "données" est obtenue directement à partir des spécifications. Il faut simplement convertir les pavés et menus édités en forme compilable. La partie "programmes" est constituée des modules de programme sélectionnés parmi l'ensemble de modules prédéfinis. La génération est totalement automatisée. L'opérateur n'a pas à intervenir. On obtient à la suite de cette phase le logiciel de communication.

Au dernier niveau, le niveau d'exécution, l'utilisateur handicapé communique avec son environnement à l'aide du logiciel de communication généré.

Remarquons que les spécifications d'un communicateur sont archivées dans des fichiers. Elles sont utiles pour des modifications ultérieures, et peuvent être utilisées pour créer d'autres communicateurs.

IV.3 Editeur interactif de communicateur

L'éditeur de communicateur est un composant intégré du système de génération. Il est utilisé pour la spécification des besoins du logiciel de communication. Celle-ci se fait par construction des données du communicateur à générer : les pavés et les menus.

Nous présentons dans ce paragraphe les divers aspects de l'éditeur : le principe d'édition, les modes de travail avec les commandes correspondantes d'édition, l'interaction opérateur-éditeur, les structures des objets manipulés et la structure du programme de l'édition.

IV.3.1 Principe de l'édition de communicateur

Comme nous l'avons étudié auparavant, les données d'un communicateur consistent en des menus et des pavés structurés. Les pavés sont les plus petits éléments affichables et sélectionnables utilisés dans la communication assistée. Les menus représentent chacun une page de pavés. La syntaxe des pavés et des menus est définie dans le paragraphe II.2.

L'édition consiste à construire "pavés et menus" suivant leur syntaxe respective. Le schéma de la figure 4.2 illustre la démarche de l'édition.

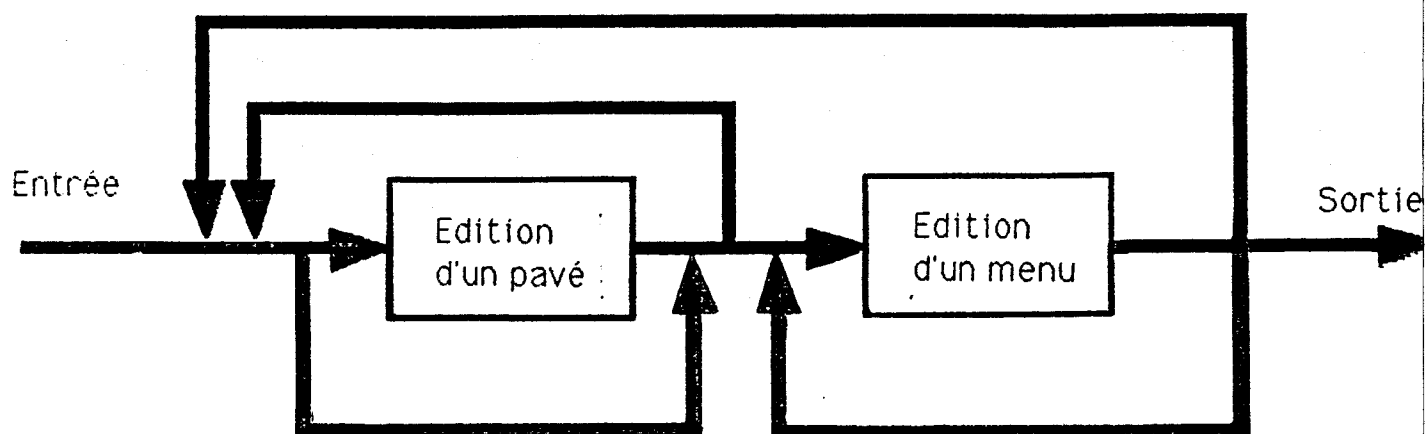


Figure 4.2 Processus d'édition de pavés et menus

La construction d'un pavé se fait par spécification de sa fonction et de son contenu. La fonction est choisie parmi une liste des fonctions disponibles proposées par l'éditeur. Le contenu, qui est sur l'écran un rectangle rempli de motifs alphanumériques ou graphiques, est spécifié par une interaction poussée entre l'éditeur et l'opérateur.

La construction d'un menu consiste à spécifier la méthode de sélection et le mode de désignation associées au menu, à sélectionner parmi les pavés construits les pavés utilisés dans le menu, et à associer à chacun de ces pavés une position dans la fenêtre d'affichage du menu. Elle se fait dans un environnement interactif.

IV.3.2 Modes de travail et commandes

La notion de "mode" est utilisée par l'éditeur de communicateur. Les différentes tâches de l'édition et les commandes d'édition correspondantes sont réparties dans quatre modes de travail :

- le mode principal,
- le mode menu,
- le mode pavé alphanumérique
- le mode pavé graphique.

La figure 4.3 illustre à l'aide d'un graphe orienté les quatre modes et les relations entre eux.

1) Mode principal

Le mode principal est le mode d'entrée et de sortie de l'édition, à partir duquel les trois autres modes sont accessibles. Les objets manipulés dans ce mode sont les pavés et les menus déjà construits. Les commandes disponibles sont les suivantes :

- créer un nouveau pavé : cette commande permet la construction d'un nouveau pavé, et elle mène à l'un des modes pavé selon le type du pavé à créer;
- créer un nouveau menu : cette commande permet de créer un menu vide, et de passer au mode menu;
- éditer un ancien menu : cette commande permet l'édition de l'un des menus déjà créés, et elle mène au mode menu;
- faire défiler en pas à pas dans les deux sens l'ensemble des pavés construits;
- supprimer le pavé visualisé;
- sauvegarder les pavés construits;
- supprimer le communicateur en cours d'édition : cette commande quitte l'édition après avoir supprimé le communicateur;
- sortir de l'édition avec sauvegarde.

La figure 4.4 illustre le mode principal avec toutes ses commandes.

2) Mode menu

En mode principal, les deux commandes "créer un nouveau menu" et "éditer un ancien menu" permettent d'accéder au mode menu. Dans ce dernier, la tâche consiste à éditer le menu visualisé dans une fenêtre d'affichage. L'opérateur peut sélectionner des pavés et les mettre dans le menu, et spécifier une méthode de sélection et un mode de désignation. Pour cela, il dispose des commandes suivantes :

- éditer un pavé dans le menu : cette commande sélectionne le pavé affiché issu de l'ensemble des pavés construits, et le met dans le menu à la position du curseur;
- déplacer le curseur : cinq touches (--> <-- ↑ ↓ et Return) permettent de déplacer le curseur dans la fenêtre d'affichage de menu;
- faire défiler en pas à pas dans les deux sens l'ensemble des pavés construits;

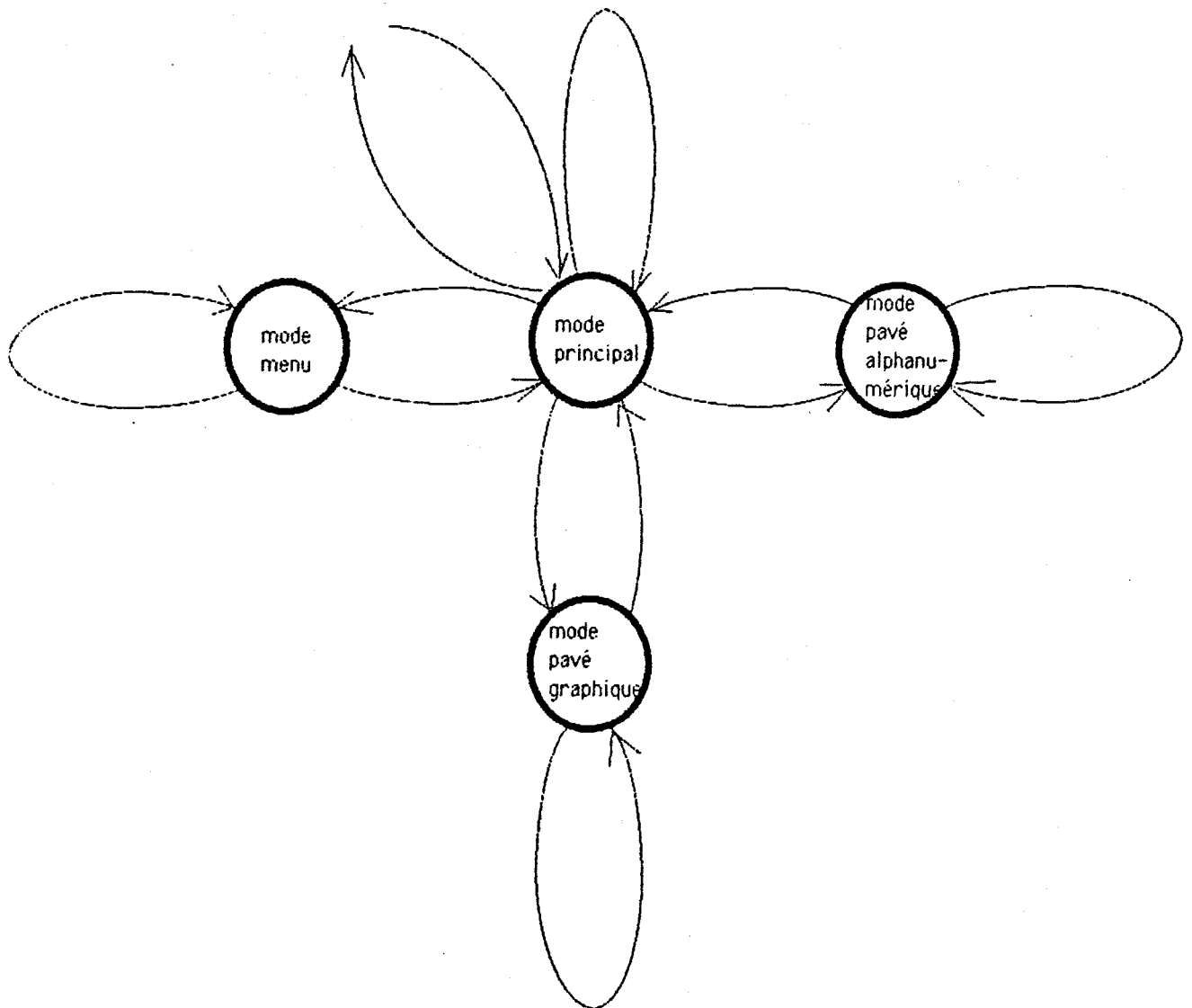


Figure 4.3 Les quatre modes de travail et les relations entre eux

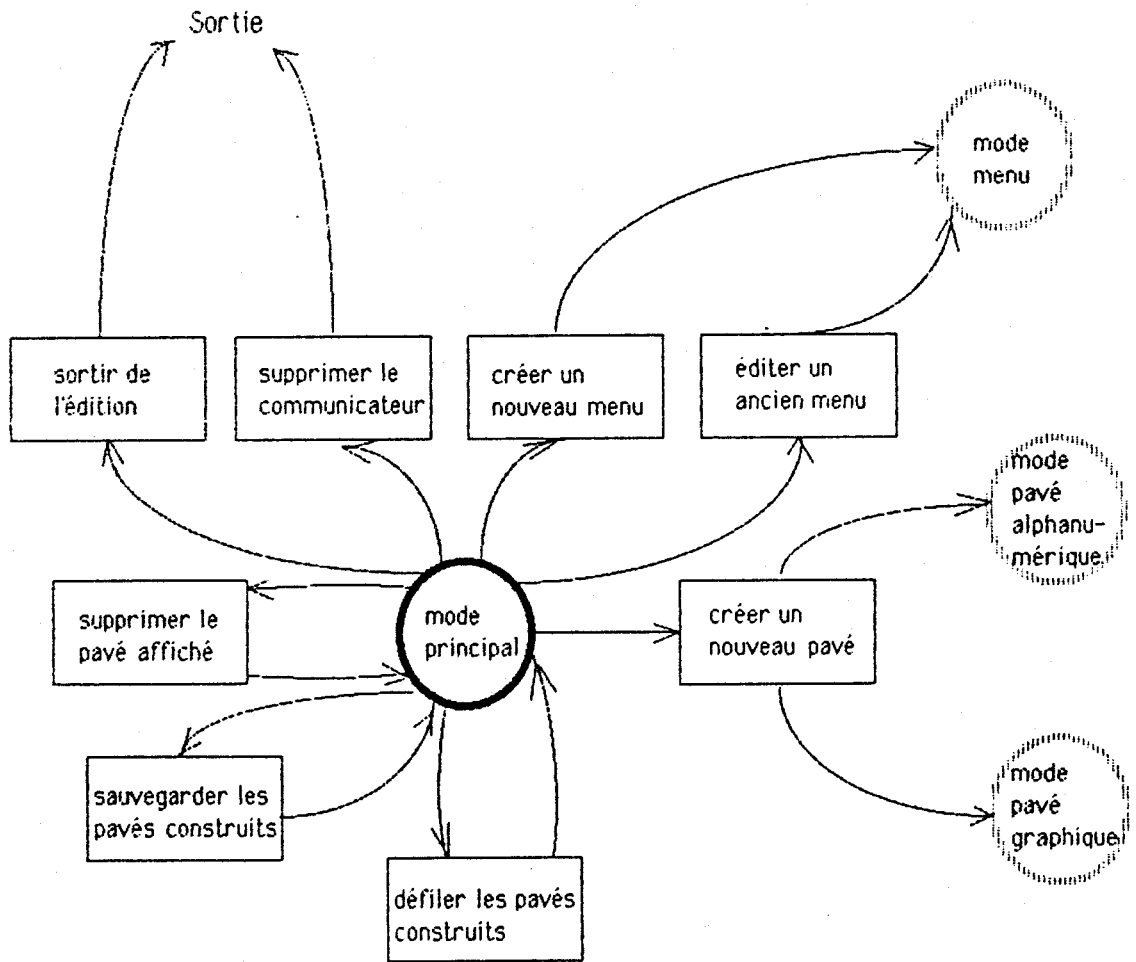


Figure 4.4 Commandes en mode principal

- déplacer le marqueur du pavé courant dans le menu : le pavé courant du menu est le pavé marqué (affiché en inverse-vidéo), deux commandes permettent sa sélection en déplaçant le marqueur dans les deux sens;

- supprimer du menu le pavé marqué;

- sauvegarder le menu;

- examiner les paramètres globaux : cette commande permet à l'opérateur d'examiner les paramètres : méthode de sélection, mode de désignation et type texte/graphique, et éventuellement de les modifier;

- supprimer du communicateur le menu en cours d'édition : cette commande supprime du communicateur le menu en cours d'édition et retourne au mode principal;

- retourner au mode principal avec sauvegarde;

- retourner au mode principal sans sauvegarde.

La figure 4.5 illustre le mode menu et toutes ses commandes.



3) Mode pavé alphanumérique

Depuis le mode principal, la commande "créer un nouveau pavé" permet d'accéder au mode pavé alphanumérique si l'opérateur choisit l'option "alphanumérique" pour le type du pavé à créer.

En mode pavé alphanumérique, l'opérateur édite le contenu du pavé en construction en le remplissant de caractères alphanumériques. La figure 4.6 présente le mode et ses commandes. Celles-ci sont :

- déplacer le curseur : cinq touches (---> <--- ↑ ↓ et Return) permettent de déplacer le curseur de caractère à l'intérieur du pavé;

- effacer le caractère sous le curseur;

- régler la dimension du pavé : l'opérateur peut régler la hauteur et la largeur du pavé façon incrémentale, l'incrément est le caractère;

- retourner au mode principal avec abandon du pavé;

- retourner au mode principal avec sauvegarde : l'opérateur est demandé de spécifier la fonction qu'il veut associer au pavé;

- écrire un caractère dans le pavé : nous considérons l'écriture d'un caractère dans le pavé comme une commande, la frappe d'une touche quelconque représentant un caractère ASCII affichable fait appel à cette commande.

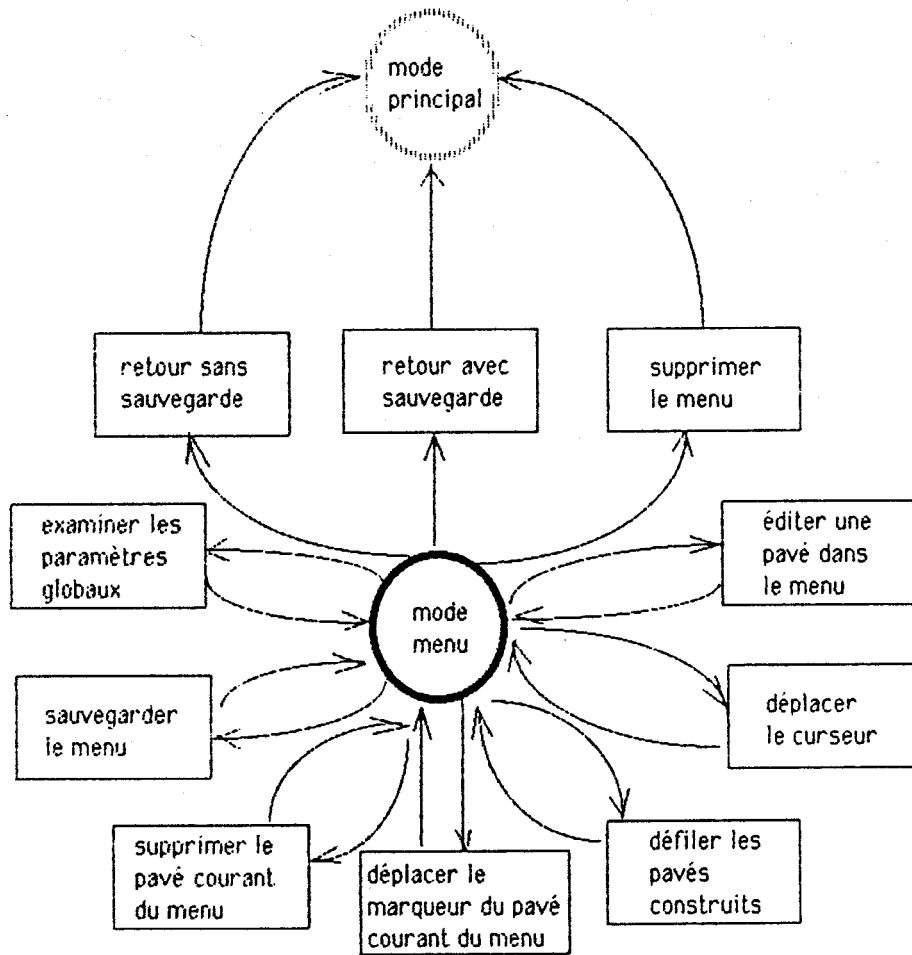


Figure 4.5 Commandes en mode menu

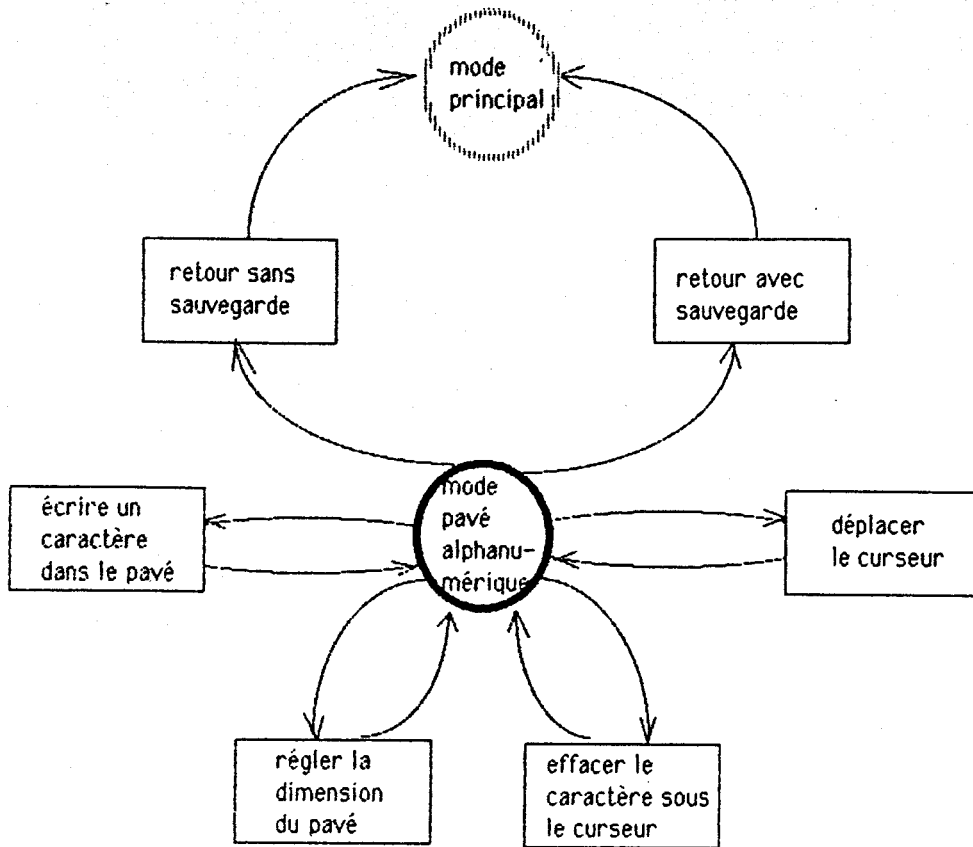


Figure 4.6 Commandes en mode pavé alphanumérique

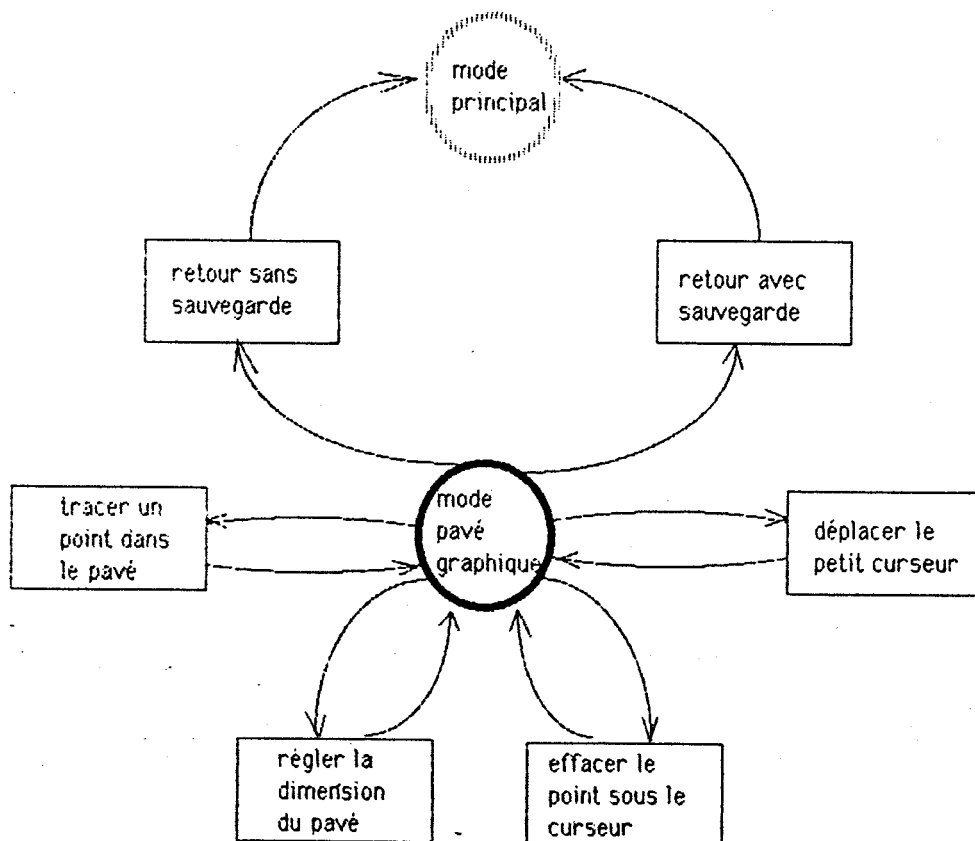


Figure 4.7 Commandes en mode pavé graphique

4) Mode pavé graphique

A partir du mode principal, la commande "créer un nouveau pavé" permet d'accéder au mode pavé alphanumérique si l'opérateur choisit l'option "graphique" pour le type du pavé à créer.

En mode pavé graphique, l'opérateur "dessine" le pavé à construire à l'aide de commandes de traçage. La figure 4.7 montre le mode pavé graphique et ses commandes. Celles-ci sont les suivantes :

- déplacer le curseur : cinq touches (--> <-- ↑ ↓ et Return) permettent de déplacer le curseur de pixel à l'intérieur du pavé;
- effacer le point sous le curseur de pixel;
- régler la dimension du pavé : la dimension du pavé se règle de manière incrémentale, l'incrément est le pixel;
- retourner au mode principal avec abandon du pavé;
- retourner au mode principal avec sauvegarde : on demande à l'opérateur de spécifier la fonction à associer au pavé;
- tracer un point : il s'agit des commandes permettant de tracer un point et de positionner le curseur de pixel.

IV.3.3 Interaction opérateur-éditeur

L'interaction opérateur-éditeur est assurée grâce au clavier et à l'écran vidéo de l'ordinateur. L'opérateur introduit ses commandes et ses données par le clavier, et l'éditeur visualise, sur l'écran, les objets manipulés, les commandes et les informations ou renseignements.

Le système de génération est destiné à être manipulé par un utilisateur non informaticien, mais connaissant le domaine de l'aide à la communication pour handicapés et capables de formuler un communicateur sous forme de pavés et de menus. Ceci exige une interface utilisateur qui cache au maximum à l'opérateur les détails et les décisions informatiques et qui l'aide au maximum dans la manipulation. C'est ainsi qu'une haute interactivité s'impose au système.

Des efforts ont été consacrés et des attentions particulières portées à la conception et la réalisation de l'interface utilisateur du système, dans le but de rendre sa manipulation aussi aisée que possible. Voici les caractéristiques importantes de l'interface :

- Inexistence de langage de commande :

Toutes les commandes d'édition sont appelées par de simples touches, et il n'existe pas de langage de commande pour l'édition. L'opérateur n'a pas à apprendre à priori un langage avant de pouvoir manipuler le système.

- Guide permanent :

Tout au long de l'édition, l'opérateur est guidé en permanence par le système : les commandes sont dynamiquement visualisées suivant le contexte de travail courant; des messages sont affichés pour aider l'opérateur dans le choix des paramètres; la cohérence entre les paramètres spécifiés est automatiquement vérifiée par le système de sorte que les objets manipulés soient toujours "corrects".

- Affichage multi-fenêtré :

L'écran vidéo est partagé en plusieurs fenêtres d'affichage qui permettent de visualiser en même temps des informations de natures différentes : objets manipulés (pavés et menus), commandes, questions et messages.

- Hiérarchie des questions et des options

Les paramètres sont spécifiés à l'aide d'un questionnaire. Pour toute question, les réponses possibles (ou options) sont visualisées. Les questions et les options sont organisées en structure arborescente de manière à ce que la quantité des informations présentée à l'opérateur soit toujours acceptable.

- "L'opérateur voit ce qu'il obtient" :

En édition, le résultat de toute action est immédiatement visible sur l'écran. De plus, les objets manipulés (pavés et menus) sont présentés à l'opérateur tels qu'ils seront dans le système cible. L'opérateur voit donc ce qu'il obtient réellement à la suite de toute action.

- Facilité de spécification des paramètres :

La spécification de certains paramètres est rendue facile et naturelle par des méthodes appropriées :

- pour associer une position à un pavé dans l'édition de menu, l'opérateur n'a pas à spécifier les coordonnées numériques (X et Y) correspondant à la position; il positionne le curseur à l'endroit voulu et la saisie des coordonnées est prise en charge par le système.

- pour spécifier la dimension du pavé en cours de construction, au lieu d'introduire par le clavier la largeur et la hauteur du pavé, l'opérateur "règle" de manière incrémentale la dimension en agissant sur de simples commandes, et il peut ainsi voir la dimension réelle du pavé.

- Sauvegarde du travail :

Des commandes permettent à l'opérateur de sauvegarder des résultats intermédiaires de l'édition. Il peut donc arrêter et reprendre son travail à son gré.

Des exemples d'utilisation du système sont présentés plus loin. Ils peuvent illustrer l'interface utilisateur dont nous parlons ici.

IV.3.4 Représentation des objets manipulés et structure du programme

IV.3.4.1 Représentation des objets manipulés

Dans l'édition d'un communicateur, les objets structurés que l'éditeur manipule sont les pavés et les menus du communicateur.

Les pavés ont la structure présentée au paragraphe II.2.1. Le paramètre "code opératoire" qui est inconnu au stade de l'édition; il est représenté par le numéro de la fonction correspondante. En mémoire, les pavés sont reliés dans une structure de liste doublement chaînée. Celle-ci facilite les opérations de manipulation telles que suppression, insertion et balayage dans les deux sens. Une procédure de compactage permet de ranger une liste dans une zone de mémoires consécutives pour qu'elle puisse être sauvegardée sur disquette.

Les menus ont la structure présentée au paragraphe II.2.2, toutefois :

- le paramètre "méthode de sélection" est représenté par le numéro de la méthode de sélection correspondante;
- les clés des pavés utilisés dans un menu sont, au lieu d'être des adresses de pavés, des numéros de pavés dans la liste de pavés construits;
- les paramètres supplémentaires sont absents.

Sur disquette, chaque menu occupe un fichier. En mémoire, il n'existe qu'un seul menu à la fois. Les noms de tous les menus d'un communicateur sont reliés dans une liste chaînée en mémoire et sauvegardés sous forme compactée dans un fichier.

On ne peut éditer qu'un seul communicateur à la fois. Les noms de tous les communicateurs existants sont sauvegardés dans un fichier et chargés en mémoire dans une liste chaînée.

IV.3.4.2 Structure du programme

L'édition se fait suivant la boucle : lecture d'une entrée du clavier, interprétation de l'entrée en fonction du contexte courant et exécution de la commande ou fonction correspondante.

L'interprétation de l'entrée se fait à l'aide d'une procédure de recherche dans une table où se trouvent les adresses de toutes les commandes ou fonctions.

L'exécution de la commande correspondante détermine le contexte et l'état suivants de l'édition.

Les commandes sont représentées par des modules de programme indépendants les uns des autres.

Cette structure rend facile la modification, l'adjonction ou la suppression de commandes.

IV.4 Générateur de communicateur

Le générateur est un programme qui permet de construire un logiciel de communication en fonction des spécifications acquises dans la phase d'édition, et à partir de l'ensemble de modules de programme prédéfinis. La génération se fait en trois étapes : génération des données, construction du programme, compilation et mise en forme du logiciel. Elle est entièrement automatisée.

IV.4.1 Génération des données de communication

Les données issues de l'édition, menus et pavés, se trouvent sous une forme non directement implémentable. Il faut leur faire subir une conversion. C'est la tâche que le générateur accomplit dans cette étape.

Il s'agit essentiellement de convertir les données de la représentation binaire en représentation d'ASCII, d'ajouter des paramètres (pour les menus par exemple) et d'insérer des instructions ou des noms de procédures d'aide à l'implémentation.

Les données converties sont rangées dans un fichier : le fichier des données à compiler.

De plus, les informations nécessaires à la sélection de modules de programme sont extraites des données et rangées dans un fichier d'informations de décision, qui est utilisé dans l'étape suivante.

Les pavés non utilisés par les menus ne sont pas inclus dans les données converties.

IV.4.2 Construction du programme de communication

La partie exécutable d'un logiciel de communication est constituée des modules paramétrés et sélectionnés parmi l'ensemble de modules prédéfinis du logiciel. Dans cet ensemble, il y a les modules intrinsèques qui sont inclus systématiquement dans le logiciel de communication, et les modules optionnels qui représentent des fonctions dépendantes des choix, telles que les méthodes de sélection, les commandes de contrôle d'édition des messages et les commandes de contrôle de l'environnement.

Le fichier d'informations de décision est utilisé dans cette étape. Il fournit d'une part les conditions de sélection des modules optionnels, et d'autre part les paramètres particuliers nécessaires pour ces modules. Pour la sélection, il s'agit d'évaluer les conditions de sélection de tous les modules en utilisant des variables booléennes dans le fichier d'informations de décision : la valeur "vrai" ou "faux" de la condition de sélection d'un module détermine l'inclusion ou exclusion du module dans le programme.

Les modules intrinsèques et les modules optionnels sélectionnés constituent le programme source du logiciel de communication.

IV.4.3 Compilation et mise en forme du logiciel de communication

Il s'agit de compiler le logiciel source de communication et de sauvegarder le résultat sous une forme appropriée directement utilisable dans le système cible. Pour les deux versions du système de génération, la démarche est différente.

Pour la version APPLE II, le compilateur croisé pour SnapForth nous permet de compiler le logiciel source sur la machine hôte APPLE II, puis de charger le code compilé dans une cartouche d'EPROM qui peut alors s'insérer dans l'un des emplacements de cartouches sur la machine cible HHC. Le logiciel est ainsi prêt à être utilisé par l'utilisateur handicapé. Au cours du développement de la version, nous avons suivi une autre démarche : les codes sources sont envoyés vers la machine cible HHC via une liaison série RS-232, et compilés sur la machine grâce à une cartouche du système SnapForth. Cette démarche nous a permis de tester facilement l'ensemble de modules de programme prédéfinis.

Pour la version IBM-PC, le logiciel source, écrit en PC/Forth+ est compilé dans le système hôte, et le résultat sauvegardé sur disquette sous la forme relogeable OVERLAY. Il peut être exécuté soit à partir de l'interpréteur du PC/Forth+, soit à partir du MSDOS au démarrage.

IV.5 Tests et essais des communicateurs générés

Des tests et essais ont été effectués au laboratoire sur des logiciels de communication générés. Pour cela, l'organe d'accès du communicateur est simulé par le clavier de la machine cible (HHC et IBM PC), et un synthétiseur de voix sert l'organe de sortie pour les messages alphanumériques.

Le système de génération a été développé au tout début pour la configuration APPLE II-HHC. De nombreux essais ont été faits sur la machine cible HHC, et ils nous ont permis de justifier le bien fondé de la structure universelle du communicateur pour handicapés, et de développer les algorithmes des fonctions de base de la communication : les méthodes de sélection, les modules de visualisation de pavé et de menu, la procédure de mémorisation de message, la procédure d'affichage de message, etc.

Pour la version IBM-PC du système de génération, tous les modules développés pour la première version ont été transcrits en PC/Forth+ et testés sur un IBM AT. De plus, nous avons ajouté des modules permettant d'avoir des pavés en couleur et des pavés graphiques.

IV.6 Exemples d'utilisation du système

Pour illustrer l'utilisation du système de génération, nous nous proposons de montrer, en présentant des écrans de visualisation, les différentes étapes à suivre lors de l'édition d'un communicateur, et la communication à l'aide d'un logiciel de communication généré. Les écrans montrés sont obtenus à partir de la version IBM-PC du système.

IV.6.1 Edition de pavés et menus

Le nom du communicateur à construire doit être spécifié pour lancer l'éditeur. Supposons qu'il soit DEMO.

Ecran 1

L'écran 1 affiche le mode principal. La première fenêtre montre le nom du communicateur (DEMO), la deuxième est réservée aux pavés construits et elle est vide puisqu'au début, il n'y a aucun pavé existant. Les commandes du mode principal sont montrées dans la troisième fenêtre. Pour créer un pavé, l'opérateur appelle la commande M.

Ecran 2

Pour créer un pavé, il faut d'abord spécifier les deux paramètres : type et couleur. L'écran 2 visualise la question " Type du pavé?". L'opérateur a le choix entre le type alphanumérique et le type graphique. Supposons que le pavé à construire soit le mot MANGER. L'opérateur appuie alors sur la touche 1 pour sélectionner le type alphanumérique.

Ecran 3

Sur l'écran 3, les 15 petits rectangles dans la fenêtre en bas représentent 15 couleurs. L'opérateur sélectionne une couleur en frappant la touche correspondante, par exemple la touche 2 pour la couleur verte.

Ecran 4

Après la sélection de couleur, le contrôle passe au mode pavé alphanumérique, le type du pavé étant alphanumérique. La fenêtre du bas visualise toutes les commandes accessibles dans ce mode (Ecran 4). Le petit rectangle dans la fenêtre au dessus représente le pavé en construction que l'opérateur peut remplir de caractères alphanumériques. Il y a un curseur dans le rectangle. Sa position est contrôlée par les touches --> <-- ↑ ↓ et Return. Le rectangle est initialement de dimension 1x1. L'opérateur doit appeler la commande F1 pour l'élargir afin de pouvoir y écrire le mot MANGER.

Ecran 1 : Début de l'edition

Communicateur: DEMO	
Commandes:	<p> N Création d'un nouveau menu O Appel d'un menu édité M Création d'un nouveau pavé R Sauvegarde des pavés construits ~X Suppression du communicateur S et T Défilement de pavés édités Del Suppression du pavé Esc Exit avec sauvegarde </p>

Ecran 3 : Spécification de la couleur de pavé

Communicateur: DEMO							
Choisissez une couleur :							
1 <input checked="" type="checkbox"/>	2 <input type="checkbox"/>	3 <input type="checkbox"/>	4 <input checked="" type="checkbox"/>	5 <input type="checkbox"/>	6 <input type="checkbox"/>	7 <input type="checkbox"/>	8 <input type="checkbox"/>
9 <input type="checkbox"/>	A <input type="checkbox"/>	B <input type="checkbox"/>	C <input type="checkbox"/>	D <input type="checkbox"/>	E <input type="checkbox"/>	F <input type="checkbox"/>	

Ecran 2 : Spécification du type de pavé

Communicateur: DEMO	
Type du pavé ?	<p> 1>Alphanumérique 2>Graphique </p>

Ecran 4 : Mode pavé alphanumérique

Communicateur: DEMO			
<input type="checkbox"/>			
Commandes en mode pavé alphanumérique :			
Esc	Exit avec sauvegarde	~E	Exit sans sauvegarde
F1	Réglage de la taille du pavé	Del	Suppression de caractère
↑	←- Enter	→	Contrôle du curseur



Ecran 5

Le réglage de la dimension du pavé se fait de façon interactive. L'opérateur dispose des quatre touches de commande : 1 2 3 et 4 qui permettent de diminuer et d'augmenter en pas à pas la hauteur ou la largeur du pavé. L'opérateur voit le changement de la dimension du pavé. Il règle ainsi la largeur du pavé à 7 caractères, et frappe la touche Esc pour retourner au mode pavé où il était.

Ecran 6

L'opérateur se retrouve en mode pavé alphanumérique (Ecran 6). Il écrit dans le pavé le mot MANGER (avec un espace derrière) à l'aide de touches alphanumériques. La spécification du contenu du pavé se termine. L'opérateur appelle la commande Esc pour passer à l'étape suivante : la spécification de la fonction à associer au pavé.

Ecran 7


La spécification de la fonction à associer au pavé se fait par sélection de l'une des fonctions prédéfinies. Le système présente à l'opérateur la liste des fonction (Ecran 7). Celui-ci n'a qu'à en sélectionner une. En l'occurrence, il sélectionne la fonction "élément informatif de base".

Pour éviter de présenter à l'opérateur trop d'informations à la fois, les fonctions sont groupées en classes et sous-classes arborescentes, et visualisées par cinq ou six.

Ecran 8

Après la spécification de la fonction du pavé, le contrôle retourne au mode principal (Ecran 8). Le pavé MANGER est sauvegardé dans la liste de pavés construits et visualisé dans la fenêtre de pavé. Il s'agit du premier pavé construit, son numéro est 1.

Ecran 9

Si l'opérateur veut construire un pavé graphique, le symbole BLISS "  " par exemple, il appelle la commande M depuis le mode principal, et sélectionne le type graphique (Ecran 2) et une couleur (Ecran3). Cela lui permet d'entrer dans le mode pavé graphique. L'écran 9 présente le pavé en construction. Les commandes disponibles sont visibles dans la fenêtres du bas. L'opérateur peut positionner le curseur de pixel, régler la dimension du pavé de la même manière qu'à l'écran 5, et dessiner son pavé à l'aide des commandes de traçage. Pour terminer la construction du pavé, il appelle la commande Esc. Comme en mode alphanumérique, il doit sélectionner une fonction pour le pavé (Ecran 7).

Ecran 5 : Réglage de la dimension de pavé

Communicateur: DEMO

Quelle est la dimension du pavé ?

Hauteur == 1 caractères	Largeur == 7 caractères
1 Plus haut	2 Moins haut
3 Plus large	4 Moins large

Esc: OK

Ecran 6 : Ecriture du contenu de pavé

Communicateur: DEMO

MANGER

Commandes en mode pavé alphanumérique :

Esc	Exit avec sauvegarde	^E	Exit sans sauvegarde
F1	Réglage de la taille du pavé	Del	Suppression de caractère
↑ ↓	← Enter	→	Contrôle du curseur



Ecran 7 : Spécification de la fonction de pavé

Communicateur: DEMO

MANGER

Fonction du pavé ?

- 1>Elément informatif de base
- 2>Commandes d'édition
- 3>Emission de messages
- 4>Contrôle de l'environnement
- 5>Changement explicite de menu

Commandes en mode pavé alphanumérique :

Esc	Exit avec sauvegarde	^E	Exit sans sauvegarde
F1	Réglage de la taille du pavé	Del	Suppression de caractère
↑ ↓	← Enter	→	Contrôle du curseur

Ecran 8 : Mode principal avec le pavé MANGER

Communicateur: DEMO

NO:1

MANGER

Commandes:

N	Création d'un nouveau menu
O	Appel d'un menu édité
M	Création d'un nouveau pavé
R	Sauvegarde des pavés construits
-X	Suppression du communicateur
S et T	Défilement de pavés édités
Del	Suppression du pavé
Esc	Exit avec sauvegarde

Ecran 10

L'opérateur a terminé la construction du pavé " 人 ". Il se retrouve en mode principal avec ce pavé affiché dans la fenêtre de pavé (Ecran 10).

Ecran 11

En mode principal, la commande N permet de créer un nouveau menu et la commande O d'éditer un menu existant. Pour créer un nouveau menu, l'opérateur doit spécifier le nom du menu (ESSAI par exemple), son type alphanumérique/graphique, sa méthode de sélection et son mode de désignation. L'écran 11 montre la façon de spécifier la méthode de sélection. L'opérateur choisit une méthode parmi la liste des méthodes disponibles, la méthode de sélection par balayage ligne-colonne par exemple.

Ecran 12

Après avoir introduit les paramètres globaux du menu ESSAI, l'opérateur se trouve en mode menu (Ecran 12). Le nom du menu ESSAI est affiché à la première ligne sur l'écran. Les commandes disponibles dans la fenêtre du bas. La fenêtre du milieu représente le cadre réservé au menu. Dans la fenêtre, il y a un curseur qui est déplaçable à l'aide des commandes --> <-- ↑↓ et Return. L'opérateur peut faire défiler la liste de pavés existants à l'aide des commandes S et T, positionner le curseur, et mettre le pavé courant de la liste dans le menu à la position du curseur. Les commandes M N et Del permettent de manipuler les pavés édités dans le menu.

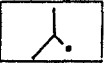
Ecran 13

A partir de l'écran 13, l'appel à la commande Y a pour but de mettre le pavé MANGER dans le menu. Le pavé est alors affiché à la position du curseur dans le menu, et devient un élément du menu. Notons que le contrôle de la cohérence entre le menu et le pavé est assuré par le système. Les deux commandes Esc et ^E permettent de terminer l'édition du menu avec ou sans sauvegarde et de retourner au mode principal.



Ecran 14

En mode menu, la commande P permet de visualiser les paramètres globaux du menu : son type alphanumérique/graphique, sa méthode de sélection et son mode de désignation, et de les modifier (Ecran 14).

Ecran 9 : Mode pavé graphique

Communicateur: DEMO			
NO:1 MANGER			
			
Commandes en mode pavé graphique :			
Esc	Exit avec sauvegarde	^E	Exit sans sauvegarde
Del	Effacement de point	T	Effacement du pavé
F1	Réglage de la taille du pavé		
M	Tracer vers la droite	N	Tracer vers la gauche
O	Tracer vers le haut	P	Tracer vers le bas
↑ ↓ → ←	Enter Contrôle du curseur		

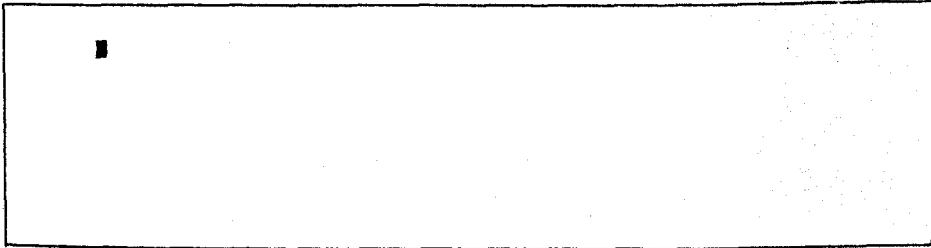
Ecran 10 : Mode principal avec le pavé graphique "  "

Communicateur: DEMO	
NO:2	
	
Commandes:	
N	Création d'un nouveau menu
O	Appel d'un menu édité
M	Création d'un nouveau pavé
R	Sauvegarde des pavés construits
^X	Suppression du communicateur
S et T	Défilement de pavés édité
Del	Suppression du pavé
Esc	Exit avec sauvegarde
	

Ecran 11 : Specification de la méthode de sélection

Communicateur: DEMO	
NO:1 MANGER	
Méthode de sélection ?	
1>Balayage pavé	2>Balayage L-C/C-L
3>Balayage dichoto	4>Codage L-C/C-L
5>Codage dichoto	5>Sans codage

Ecran 12 : Mode menu

Communicateur: DEMO	Menu: ESSAI		
NO:1 MANGER			
			
Commandes en mode menu:			
Y	Edition du pavé courant dans le menu	M N	Déplacement du marqueur
S T	Défilement des pavés construits	R	Sauvegarde du menu
^X	Suppression du menu	Del	Suppression du pavé marqué
Esc	Exit avec sauvegarde	^E	Exit sans sauvegarde
↑ ↓ → ←	Enter Contrôle du curseur		
		P	Paramètres du menu

Ecran 13 : Edition d'un pavé dans le menu

Communicateur: DEMO Menu: ESSAI

NO:1
MANGER

MANGER

Commandes en mode menu:

Y Edition du pavé courant dans le menu M N Déplacement du marqueur
 S T Défilement des pavés construits R Sauvegarde du menu
 ~X Suppression du menu Del
 Esc Exit avec sauvegarde ~E Exit sans sauvegarde
 ^ v -> <- Enter Contrôle du curseur P Paramètres du menu

Ecran 15 : Menu "commandes"

Communicateur: DEMO Menu: COMMANDES

NO:1
droite

droite gauche init-curseur efface-pavé efface-message
 autre-buffer autre-fenêtre insert/érase exit
 >synthétiseur
 ~VERBES >BOISSONS >PRINCIPAL >LETTRES

Commandes en mode menu:

Y Edition du pavé courant dans le menu M N Déplacement du marqueur
 S T Défilement des pavés construits R Sauvegarde du menu
 ~X Suppression du menu Del
 Esc Exit avec sauvegarde ~E Exit sans sauvegarde
 ^ v -> <- Enter Contrôle du curseur P Paramètres du menu

Ecran 14 : Examenation des paramètres du menu

Communicateur: DEMO Menu: ESSAI

NO:1
MANGER

MANGER

Type du menu == Alphanumérique
 Méthode de sélection == Balayage ligne/colonne
 Mode de désignation == Inversion vidéo
 Esc> Exit

Ecran 16 : Menu "alphabet"

Communicateur: DEMO Menu: LETTRES

NO:1
droite

d	e	j	a	q	n	o	7
p	c	m	o	f	i	B	
s	t	r	b	z	9		
u	i	j	3				
v	h	4	y				
g	5	w	>COMMANDES				
6	x	>PRINCIPAL					

Commandes en mode menu:

Y Edition du pavé courant dans le menu M N Déplacement du marqueur
 S T Défilement des pavés construits R Sauvegarde du menu
 ~X Suppression du menu Del
 Esc Exit avec sauvegarde ~E Exit sans sauvegarde
 ^ v -> <- Enter Contrôle du curseur P Paramètres du menu



Ecrans 15 et 16

A titre d'exemple, l'écran 15 montre un menu "commandes" où on trouve des commandes de contrôle d'édition des messages, et les éléments >VERBES >BOISSONS, >PRINCIPAL et >LETTRES permettant la sélection de divers menus par l'utilisateur.

L'écran 16 présente un menu "alphabet" [TIC. 85] où on trouve les lettres et les chiffres. Les éléments du menu sont rangés suivant l'ordre décroissant de leurs fréquences d'utilisation de façon à augmenter la vitesse de la construction du message.

IV.6.2 Communication

Nous illustrons la communication à l'aide d'un logiciel de communication généré par notre système de génération. Il y a cinq menus dans le communicateur : le menu principal, le menu "commandes", le menu "alphabet", le menu "verbes" contenant quelques verbes et le menu "boissons" présentant les noms de quelques boissons.

Ecran 17

La communication commence par le menu principal où se trouvent les pavés permettant d'accéder à un autre menu (Ecran 17). Les deux fenêtres occupant la partie basse de l'écran sont réservées à l'affichage du messages primaire. L'utilisateur ne travaille qu'avec une seule fenêtre à la fois.

Supposons que l'opérateur veuille construire le message je veux boire du jus d'orange. Il sélectionne d'abord le pavé >VERBES à l'aide de la méthode de sélection par balayage en série, ce qui lui permet de visualiser le menu "verbes".

Ecrans 18 et 19

Le menu en cours propose quelques verbes et des pavés de changement de menu (Ecran 18). La fenêtre "Message 1" est utilisée pour l'affichage du message et elle est initialement vide. Par balayage en série, l'opérateur sélectionne d'abord le pavé Je veux, puis le pavé boire. Ces deux pavés représentent des éléments informatifs de base, ils se visualisent alors dans la fenêtre "Message 1" (Ecran 19). Pour visualiser le menu "boissons", l'opérateur choisit le pavé >BOISSONS.

Ecran 20

Le menu en cours présente les noms de quelques boissons (Ecran 20). L'opérateur sélectionne le pavé jus d'orange. Celui-ci s'affiche alors dans le message. A ce moment, l'opérateur s'aperçoit qu'il a oublié de mettre l'article du avant le pavé jus d'orange, il peut sélectionner le pavé >COMMANDES pour appeler le menu "commandes".

Ecran 17 : Menu principal

Méthode de sélection : Balayage en série

>VERBES >BOISSONS >LETTRES

>COMMANDES exit

Message 1

Message 2

Ecran 19 : Sélection de pavés dans le menu "verbes"

Méthode de sélection : Balayage en série

boire manger écouter regarder

Je veux

>COMMANDES >PRINCIPAL >LETTRES

Message 1

Je veux boire

Message 2

Ecran 18 : Menu "verbes"

Méthode de sélection : Balayage en série


boire manger écouter regarder

Je veux

>COMMANDES >PRINCIPAL >LETTRES

Message 1

Message 2



Ecran 20 : Menu "boissons"

Méthode de sélection : Balayage en série

cocacola jus d'orange vin

cidre champagne

>COMMANDES >PRINCIPAL >LETTRES

Message 1

Je veux boire jus d'orange

Message 2

Ecran 21

Le menu "commandes" propose des éléments de commandes de contrôle d'édition des messages. L'opérateur sélectionne le pavé gauche, dont la fonction est de positionner le curseur sur le pavé à gauche de sa position actuelle, en l'occurrence sur le pavé jus d'orange (Ecran 21). Ensuite, Il sélectionne le pavé >LETTRES pour visualiser le menu "alphabet".

Ecrans 22 et 23

Le menu "alphabet" présente les lettres et les chiffres. Par balayage ligne-colonne, l'opérateur peut sélectionner la lettre d et la lettre u pour composer l'article du. Ceci complète le message Je veux boire du jus d'orange (Ecran 22). Puis il peut visualiser le menu "commandes", et sélectionner le pavé >SYNTHETISEUR pour faire parler la phrase Je veux boire du jus d'orange (Ecran 23).

Ecran 21 : Sélection d'une commande d'édition

Méthode de sélection : Balayage ligne-colonne

droite	gauche	init-curseur	efface-pavé	efface-message
autre-buffer	autre-fenêtre	insert/écrase	exit	
>synthétiseur				
>VERBES	>BOISSONS	>PRINCIPAL	>LETTRES	

Message 1
Je veux boire jus d'orange []
↑

Message 2
[]

Ecran 23 : Menu "commandes" et le message
Je veux boire du jus d'orange

Méthode de sélection : Balayage ligne-colonne

droite	gauche	init-curseur	efface-pavé	efface-message
autre-buffer	autre-fenêtre	insert/écrase	exit	
>synthétiseur				
>VERBES	>BOISSONS	>PRINCIPAL	>LETTRES	

Message 1
Je veux boire du jus d'orange

Message 2
[]

Ecran 22 : Menu "alphabet" et le message
Je veux boire du jus d'orange

Méthode de sélection : Balayage ligne/colonne

d	e	l	a	q	n	0	?
p	c	m	o	f	1	8	
s	t	r	b	:	9		
u	i	j	3				
v	h	4					
g	5	w	>COMMANDES				
6	x	>PRINCIPAL					

Message 1
Je veux boire du jus d'orange

Message 2
[]



Conclusion

La génération automatique de logiciels d'aide à la communication pour handicapés a été concrétisée par les différentes implémentations réalisées : une version du système de génération pour APPLE II, une autre, plus puissante, pour IBM PC ou compatible, et des communicateurs générés pour des tests.

Cette concrétisation permet de prouver la possibilité de générer automatiquement des logiciels de communication à partir de spécifications simples ne reflétant que des concepts du domaine de l'aide à la communication pour handicapés, la validité de la structure générale universelle du communicateur et la portabilité du système de génération vis à vis de la machine hôte et de la machine cible.

CONCLUSION GENERALE

Un système d'aide à la conception de logiciels de communication pour handicapés a été présenté dans ce mémoire. Il permet la génération automatique de logiciels de communication adaptés aux besoins spécifiques du handicapé, avec une réduction considérable en effort et temps de développement. Cela est possible grâce à la généralisation et à la structuration du système de communication.

Différentes implémentations du système dans différentes configurations ont permis de prouver la validité de la structure générale du système de communication.

Le travail présenté constitue seulement une étape dans l'informatisation des systèmes de communication à l'usage des handicapés. La phase à venir est de pouvoir appliquer notre dispositif avec des utilisateurs non informaticiens dans des centres de rééducation et pour divers cas de handicapés. Cela permettrait d'obtenir un ensemble de suggestions et d'améliorations. Le système ayant été construit de façon modulaire et dans un langage extensible ne pose aucun problème d'évolution. La seule difficulté réside dans l'écriture de nouvelles fonctions.

Pour conclure, nous pouvons citer les travaux envisageables afin de compléter notre système :

- la réalisation de méthodes d'accélération,
- l'implémentation de transformations,
- la définition de la structure du dictionnaire incorporé dans le logiciel de communication,
- la génération du dictionnaire à partir d'une base de données (un dictionnaire général standard),
- l'évolution du système vers un système expert,
- la transcription du système dans un langage plus extensible et plus favorable à la génération, le langage Smalltalk par exemple [LER. 82a].

BIBLIOGRAPHIE

- [AYL. 81]
 J.H. AYLOR, B.W. JOHNSON et R.L. RAMEY
 The Impact of Microcomputers on Devices to Aid the Handicapped
 IEEE Computer, Vol.14, No.1, Jan. 1981, PP.35-40.
- [AYL. 83]
 J.H. AYLOR et E.L. JOHNSON
 Guest Editors' Introduction : Microcomputing to Aid the Handicapped
 IEEE Micro, Vol.3, No.3, June 1983, PP.6-7.
- [BAL. 81]
 R. BALZER
 Special Issue on Program Transformations
 IEEE Trans. Soft. Eng. SE-7 (1), Jan. 1981.
- [BAL. 85]
 R. BALZER
 A 15 Years Perspective on Automatic Programming
 IEEE Trans. on Soft. Eng., SE-11 (11), Nov. 1985, pp.1257-1267.
- [BAR. 79a]
 D. BARSTOW
 An Experiment in Knowledge-Based Automatic Programming
 Artificial Intelligence 12 (1979), pp.73-119.
- [BAR. 79b]
 D. BARSTOW
 Knowledge-Based Program Construction
 North Holland, The Computer Science Library, 1979.
- [BAR. 85]
 D.R. BARSTOW
 Domain-Specific Automatic Programming
 IEEE Trans. on Soft. Eng., SE-11 (11), Nov. 1985, pp.1321-1336.
- [BEA. 83]
 M. BEAUDOUIN-LAFON, C. GRESSE et G. GUIHO
 CATY : un Environnement de programmation pour une Construction
 Graphique et Interactive de Programmes
 Conférence Invitée aux Journées Internationales EUROSOFT sur la
 Construction de Programmes, Paris, Nov. 1983.
- [BIE. 75]
 A.W. BIERMANN, R.I. BALM et F.E. PETRY
 Speeding Up the Synthesis of Programs
 IEEE Transaction on Computers, Vol.C-24 (2), 1975, pp.122-136.
- [BIE. 76]
 A.W. BIERMANN, et R. KRISHNASWAMY
 Constructing Programs from Example Computations
 IEEE Trans. Soft. Eng. SE-2 (3), Sept. 1976.
- [BLA. 81]

H.L. BLASZCZYK

Computing and the Handicapped : The Challenge in Education
IEEE Computer, Vol.14, No.1, Jan. 1981, pp.15-17.

[BOE. 82]

B.W. BOEHM

Les Facteurs du Coût de Logiciel
TSI, Vol.1, No.1, Fév. 1982, pp.5-24.

[BOU. 80]

D. BOURDACHE, A. BOURDACHE et M. LAMOTTE

Système d'Aide Visuelle pour Handicapés Auditifs
ITBM, Vol.1, No.3, 1980.

[BOU. 85]

G. BOURHIS

Conception et Automatisation d'un Désigne Polyvalent Géré par
Microprocesseur : Application aux Grands Handicapés Moteurs Présentant des
Troubles de la Communication Verbale
Thèse Docteur-Ingénieur, NANCY, I.N.P.L, Dec. 1985.

[BOU. 86]

G. BOURHIS, J.M. ANDRE et E.YVROUD

Désigne Polyvalent pour Grands Handicapés Moteurs Présentant des
Troubles de la Communication Verbale
ITBM, Vol.7, No.4, 1986, pp.474-481.

[BRA. 82]

J.A. BRABYN

Mobility Aids fo the Blinds
IEEE Engineering in Medicine and Biology Magazine
Vol.1, No.4, Dec. 1982, pp.36-38.

[BRA. 85]

J. BRABYN

Development in Electronic Aids for the Blind and Visually Impaired
IEEE Engineering in Medicine and Biology Magazine
Vol.4, No.4, Dec. 1985, pp.33-37.

[BRA. 86]

M. BRADLEY

A Window System
The Journal of Forth Application, Vol.3, No.2, 1986, pp.105-108.

[BRO. 81]

L. BRODIE

Starting FORTH
Prentice-hall, Englewood Cliffs, N.J., 1981.

[BRY. 82]

J. BRYANT

Rehabilitation Engineering : The User's Perspective
IEEE Engineering in Medicine and Biology Magazine
Vol.1, No.4, Dec. 1982, pp.39-40.

- [CHE. 84] T.T. CHENG, E.D. LOCK et N.S. PRYWES
Use of Very High Level Languages and Program Generation by Management Professionals
IEEE Trans. on Soft. Eng., SE-10(5), Sept.1984, pp.552-563.
- [CHE. 80] J.L. CHEVAL et J. MOSSIERE
Evaluation d'un Système d'Aide à la Production de Logiciel Modulaire :
Sésame
BIGRE 80.
- [CHE. 82] J.L. CHEVAL, J. ESTUBLIER, S. GHOUL et S. KRAKOWIAK
Modularité et Composition des Programmes dans l'Atelier de Logiciel ADELE
1er Congrès de Génie Logiciel, Paris, 1982, pp.183-197.
- [CHI. 82] D.CHILDRESS
Myoelectric Control of Powered Prostheses
IEEE Engineering in Medicine and Biology Magazine
Vol.1, No.4, Dec. 1982, pp.23-25.
- [COL. 78] L.S. COLES
Interactive Programming of a Medical Telem manipulator
Proceeding of the International Conference on Telem manipulators for the
Physically Handicapped,
Rocquencourt, France, Sept. 1978, pp.219-231.
- [DAV. 77] R. DAVIS, B. BUCHANAN et E. SHORTLIFFE
Production Rules as a representation for a Knowledge-based Consultation
Program
Artificial Intelligence 8 (1), Fed. 1977, pp.15-46.
- [DER. 82] F. DEREMER et H.H. KRON
Programming-in-the-Large versus Programming-in-the-Small
IEEE Trans. on Soft. Eng., SE-2 (2), June 1982, pp.80-82.
- [DER. 77] N. DERSHOWITZ et Z. MANNA
The Evolution of Programs : A Systems for Automatic program Modification
IEEE Trans. Soft. Eng. SE-3 (4), Sept. 1977.
- [DES. 86] X. DESTOOP et E. TAQUET
ACTIONCOM
Manuel d'utilisation, Centre Marc Sautet, Villeneuve d'Ascq, 1986.
- [DRO. 82] E. DE DROUAS et J.M. NERSON
Les Ateliers Logiciels Intégrés : développements Français Actuels
TSI., Vol.1, No.3, 1982.

- [DUB. 79]
J.P. DUBUS et F. WATTRELOT
Interpréteur-Editeur Braille Automatique Autonome avec Clavier de Machine à Ecrire
Le Nouvel Automatism, Tome XXIV, No.6, Juin-Juillet 1979.
- [DUB. 82]
J.P. DUBUS et F. WATTRELOT
A Study of the Visualization of Alphanumerical Text on a TV Screen for the Use of Partically Sighted Persons
Microprocessing and Microprogramming Vol.9, No.3, Mar. 1982, pp.133-141.
- [ESC. 80]
P. ESCHSTUTH
Un Appareil de Communication pour Grands Handicapés Moteurs à Base de Techniques Vidéo et Microprocesseur
ITBM, Vol.1, No.3, 1980, pp.4-12.
- [FIC. 85]
S.F. FICKAS
Automating the Transformational Development of Software
IEEE Trans. on Soft. Eng., SE-11 (11), Nov. 1985, pp.1268-1277.
- [FOI. 80]
J. FOISSEAU, R. JACQUART, M. LEMAITRE, M. LEMOINE et G. ZANON
Recherches sur la Conception de Programmes Assistée par Ordinateur
Rapport Final Convention ONERA-RGA CERT, 1980.
- [GIA. 81]
M.J.GIANNINI
Computing and the Handicapped : A Promising Alliance.
IEEE Computer, Vol.14, No.1, Jan. 1981, pp.12-13.
- [GIB. 81]
C.D. GIBLER
Linguistic and Human Performance Considerations in the Design of an Anticipatory Communication
PhD thesis, Northwestern University, Evanston, Illinois, 1981.
- [GIB. 82]
C.D. GIBLER et D.S. CHILDRESS
Language Anticipation with a Computer-Based Scanning Communication Aid
Proc. IEEE Computer Society Workshop on Computing to Aid the Handicapped, Charlooesville, Virginia, 1982, pp.11-16.
- [GOR. 80]
R.D. GORDON
The Modular Application Customizing System
IBM Syst. J. 19 (4), 1980, pp.521-541.
- [GRE. 78]
C. GREEN et D. BARSTOW
On Program Synthesis Knowledge
Artificial Intelligence 10 (3), Nov. 1978, pp.241-280.

[GRE. 76]

C. GREEN

The Design of the PSI Program Synthesis System
 Proc. 2nd International Conference on Software Engineering,
 IEEE Computer Society, Long Beach, Ca, oct.1976, pp.4-18.

[GRE. 81]

C. GRESSE

A propos de la Synthèse de Programmes à Partir de Spécifications
 Congrès AFCET INRIA Reconnaissances des formes et Intelligence
 Artificielle, Nancy, 1981.

[GRE. 82]

C. GRESSE et G. GUIHO

Proposition d'un Environnement Interactif pour Faire de la Construction
 assistée de Programmes
 1er Colloque Génie logiciel, Paris, 1982, pp.65-75.

[GRE. 84]

C. GRESSE

Contribution à la Programmation Automatique, " CATY : un System de
 Construction Assistée de Programmes "
 Thèse d'Etat, Univ. Paris-Sud, Orsay, Mars 1984.

[GRE.]

C. GRESSE et G. GUIHO

Synthèse des Programmes de Synthèse de Programmes
 LRI, Univ. Paris-Sud.

[GRO. 83]

C.P. GROSSNER T.RADHAKRISHNAN et A. POSPIECH

An Integrated Workstation for the Visually Handicapped
 IEEE Micro, Vol.3, No.3, June 1983, pp.8-16.

[GRO. 86]

C.P. GROSSNER, T. RADHAKRISHNAN et A. SCHENA

An Intelligent Braille Display Device
 IEEE Micro, Vol.6, No.3, June. 1986, pp.43-51.

[GUI. 80]

G. GUIHO, C. GRESSE et M. BIDOIT

Conception et Certification de Programmes à Partir d'une Décomposition par
 les Données
 Revue RAIRO Informatique, Vol.14, No.4, 1980, pp.319-351.

[GUI. 76]

J. GUITTET

The Spartacus Project : Aids for the Physically Handicapped
 International Congress on the Technology for Prosthetics and Functional
 Rehabilitation, Cannes, France, Mar. 1976.

[GUI. 78]

J. GUITTET H.H. KEE N. QUETIN et J. YCLON

The Spartacus Telethesis : Manipulator Control and Experimentation
 Proceeding of the International Conference on Telemanipulators for the
 Physically Handicapped,
 Rocquencourt, France, Sept. 1978, pp.79-102.

- [HAN. 85]
D.F. HANSON et P.C.POWER
Electronic Scanners with Speech Output -- A Communication System for the Physically Handicapped and Mentally Retarded
IEEE Micro, Vol.5, No.2, Apr. 1985, pp.25-52.
- [HE 84]
Bing HE
Etude d'un système d'aide à la conception et à la programmation de prééditeurs
Rapport du DEA d'Automatique et Informatique Industrielle, U.S.T.L.F.A. 1984.
- [HE 87]
Bing HE, J. TICHON et J.M. TOULOTTE
Génération de logiciels d'aide à la communication pour handicapés
ITBM, à paraître
- [HEC. 83]
C.W. HECKATHORNE et D.S. CHILDESS
Applying Anticipatory Text Selection in a Writing Aid fo People with Severe Motor Impairment
IEEE Micro, Vol.3, No.3, June 1983, pp.17-23.
- [HEW. 75]
C. E. HEWITT et B. SMITH
Towarde a Programming Apprentice
IEEE Trans. Soft. Eng. SE-1 (1), Mar. 1975, pp.26-45.
- [HON. 85]
W.M.HONIG et R.H. EIKELBOOM
Microprocessor-Based Television Games, Exercises, and Evaluation Procedures for the Physically and Mentally Handicapped
IEEE Engineering in Medicine and Biology Magazine Vol.4, No.4, Dec. 1985, pp.43-50.
- [IEE. 81]
IEEE Transaction on Software Engineering
Special Section on Programming Environments
Vol.SE-7, No.5, sept. 1981.
- [JAF. 85]
D.L. JAFFE
Forth In Rehabilitation
Forth Dimensions, Vol.6, No.2, 1985.
- [KLI. 78]
V. KLIG
Biomedical Applications of Microprocessors
Proceeding of IEEE, Vol.66, No.2, Feb.1978, pp.151-161.
- [KRA. 82]
S. KRAKOWIAK
Systèmes Intégrés de Production de Logiciel : Concepts et Réalisation
Actes des Journées BIGRE, Grenoble, Jan. 1982.

[KRA. 82]

S. KRAKOWIAK
Systèmes Intégrés de Production de Logiciels : Concepts et Réalisation
TSI, Vol.1, No.3, 1982, pp.187-200.

[LAU. 82]

J.L. LAURIERE
Représentation et Utilisation des Connaissances
TSI, Vol.1, No.1 et 2, 1982.

[LEB. 82]

M. LEBLANC
Environmental Control and Robotic Manipulation Aids
IEEE Engineering in Medicine and Biology Magazine Vol.1, No.4, Dec. 1982,
pp.16-19.

[LEN. 81]

R.J. LENEWAY et B.R. MONTGOMERY
Rehabilitation and the Handicapped Programmer
IEEE Computer, Vol.14, No.1, Jan. 1981, pp.49-52.

[LER. 82a]

E.J. LERNER
Programming for Nonprogrammers
IEEE Spectrum, Vol.19, No.8, Aug. 1982, pp.33-38.

[LER. 82b]

E.J. LERNER
Automating Programming
IEEE Spectrum, Vol.19, No.8, Aug. 1982, pp.28-32.

[LIN. 82]

M.W. LINVILL
The Economics of Placing the Disabled in the Labor Market
IEEE Engineering in Medicine and Biology Magazine Vol.1, No.4, Dec. 1982,
pp.19-22.

[LUN. 83]

D. LUNNEY, R.C. MORISON, M.M. CETERA et R.V.HARTNESS
A Microcomputer-Based Aid for Visually Impaired Students
IEEE Micro, Vol.3, No.4, Aug. 1983, pp.19-31.

[MAN. 84]

A.MANDAR et J.P. DUBUS
Nouvelle Technique de Communication à l'Usage des Infirmes Moteurs
Cérébraux par Edition de Matrice Dynamique
ITBM, Vol.5, No.6, 1984, 657-664.

[McL. 82]

C.A. McLAURIN
Wheelchairs : Technical Design
IEEE Engineering in Medicine and Biology Magazine Vol.1, No.4, Dec. 1982,
pp.28-30.

- [McN. 82]
D.R. McNEAL
Rehabilitation Engineering
IEEE Engineering in Medicine and Biology Magazine Vol.1, No.4, Dec. 1982,
pp.15-16.
- [MED. 81]
R. MEDINA-MORA et P. FEILER
An Incremental Programming Environment
IEEE Trans. on Soft. Eng., SE-7 (5), Sept. 1981, pp.472-481.
- [MIL. 79]
E. MILLER
Automated Tools for Software Engineering
Tutorial COMPSAC 79, IEEE Catalog No.150-3, pp.1-5.
- [MIL. 74]
L.A. MILLER
Programming by Non-Programmers
Int. J. Man-Machine studies 6, 1974, pp.237-260.
- [MIL. 81]
L.A. MILLER
Natural Language Programming : Styles, Strategies and Contrasts
IBM Syst. J. 20 (2), 1981, pp.184-215.
- [MYE. 82]
W. MYERS
Personal Computers Aid the Handicapped
IEEE Micro, Vol.2, No.1, Feb. 1982, pp.20-40.
- [NEL. 83]
P.J. NELSON, L.KORBA, G.PARK et D.CRABTREE
The MOD Keyboard
IEEE Micro, Vol.3, No.4, Aug. 1983, pp.7-17.
- [OUA 86]
B. OUASSIR
Techniques d'accélération pour un système d'aide à la communication
Rapport du DEA d'Automatique et Informatique Industrielle, U.S.T.L.F.A.
1986.
- [PAR. 72]
D. PARNAS
On the Criteria to Be Used in Decomposing Systems into Modules
Comm. ACM 15 (12), Dec. 1972, pp.1053-1058.
- [PAR. 79]
D. PARNAS
Designing software for Ease of extension and Contraction
IEEE Trans. Soft. Eng. SE-5 (2), Mar. 1979, pp.128-137.
- [PAR. 85]
D. PARNAS P.C. CLEMENTS et D.M. WEISS
The Modular Structure of Complex Systems
IEEE Trans. Soft. Eng. SE-11 (3), Mar. 1985, pp.259-266.

- [PEC. 84]
H. PECKHAM
Le Basic sur le bout des doigts
McGRAW-HILL Editeurs, 1984.
- [PEL.]
C. PELLEGRINI
Un Système Interactif d'Aide à la Conception de Programmes
Thèse d'Etat, Suisse.
- [QUI. 82]
V. QUINT
An Interactive System for Editing Mathematical Documents
European Conference on Integrated Interactive Computing Systems, Stresa,
Sept. 1982.
- [QUI. 83]
V. QUINT
Un Système Interactif pour la Production de Documents Mathématiques
TSI, Vol.2, No.3, 1983, pp.179-190.
- [RAH. 81]
M.A.RAHIMI
Intelligent Prosthetic Devices
IEEE Computer, Vol.14, No.1, Jan. 1981, pp.19-23.
- [RAM. 79]
R.L. RAMEY, J.H.AYLOR et R.D.WILLIAMS
Microcomputer-Aided Eating for the Severely Handicapped
IEEE Computer, Vol.12, NO.1, Jan. 1979, pp.54-61.
- [ROS. 82]
M.J. ROSEN
Communication Systems for the Nonvocal Motor Handicapped : Practice &
Prospects
IEEE Engineering in Medicine and Biology Magazine Vol.1, No.4, Dec. 1982,
pp.31-35.
- [SCH. 81]
W. SCHNEIDER, G. SCHMEISSER et W. SEAMONE
A Computer-Aided Robotic Arm/Worktable System for the High-level
Quadriplegic
IEEE Computer, Vol.14, No.1, Jan. 1981, pp.41-47.
- [SUE. 79]
C.Y. SUEN
n-Gram Statistics for Natural Language Understanding and Text Processing
IEEE Trans. on Pattern Analysis and Machine Intelligence,
Vol.PAMI-1, No.2, Apr.1979, pp.164-172.
- [THO. 81]
A. THOMAS
Communication Devices for the Nonvocal Disabled
IEEE Computer, Vol.14, No.1, Jan. 1981, pp.25-30.

- [TIC. 85]
J. TICHON
 Conception et Réalisation d'un Système de Communication pour Handicapés,
 Utilisant des Techniques d'Accès à un Dictionnaire
 Thèse d'Université, U.S.T.L.F.A, Juillet 1985.
- [TIC. 87]
J. TICHON, Bing HE et J.M. TOULOTTE
 Generation of Specific Communicators for the Handicapped Persons
 34th ISMM International Conference on Mini and Microcomputers,
 Lunago, Switzerland, July 1987.
- [TOU. 83]
Logiciel
J.M. TOULOTTE et J.M. CICHOWLAS
 La Prédiction dans l'Aide à la Communication
 HANDITEC, 1983.
- [TOU. 86]
J.M. TOULOTTE et J.TICHON
 Development of a Communication System for Handicapped Persons,
 Generation of Systems
 8th Annual Conference of IEEE Engineering in Medicine and Biology Society,
 Nov. 1986, DALLAS-FORT WORTH, U.S.A
- [TOU. 87]
J.M. TOULOTTE et J. TICHON
 Conception et Réalisation d'un Système de Communication pour Handicapés
 ITBM, à paraître.
- [VAN. 82a]
A. VAN LAMSWEERDE
 Les Outils d'Aide au Développement de Logiciels : un Aperçu des Tendances
 actuelles
 XVe Journées Internationales de l'Informatique et de l'Automatisme, Paris,
 Juin 1982.
- [VAN. 82b]
A. VAN LAMSWEERDE
 Automatisation de la Production de Logiciels d'Application : Quelques
 Approches, 1ère Partie
 TSI, Vol.1, No.6, 1982, pp.475-494.
- [VAN. 83a]
A. VAN LAMSWEERDE
 Automatisation de la Production de Logiciels d'Application : Quelques
 Approches, 2e Partie
 TSI, Vol.2, No.1, 1983, pp.5-20.
- [VAN. 83b]
A. VAN LAMSWEERDE
 Automatisation de la Production de Logiciels d'Application : Quelques
 Approches, 3e Partie
 TSI, Vol.2, No.2, 1983, pp.81-93.
- [VAN. 78]
G. VANDERHEIDEN
 Non-Vocal Communication Resource Book
 University Park Press, Baltimore, 1978.

[VAN. 81]

G.C. VANDERHEIDEN

Practical Application of Microcomputers to Aid the Handicapped
IEEE Computer, Vol.14, No.1, Jan. 1981, pp.54-61.

[WAR. 82]

G.WARREN

Weelchairs : How They Are Used
IEEE Engineering in Medicine and Biology Magazine
Vol.1, No.4, Dec. 1982, pp.26-28.

[WAT. 82]

R. WATERS

The Programmers' Apprentice : Knowledge-Based Program Editing
IEEE Trans. on Soft. Eng., SE-8 (1), Jan. 1982.

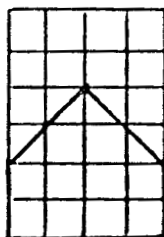
ANNEXES

A.1. LE BLISS [TIC. 85]

Le BLISS est un moyen de communication pour des sourds-muets et des handicapés mentaux ne pouvant assimiler l'écriture arabe.

Il est basé sur l'emploi de symboles pictographiques, idéographiques, internationaux ou arbitraires.

a) Exemples de symboles pictographiques.

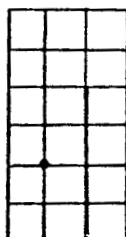


maison



homme

b) Exemples de symboles idéographiques.



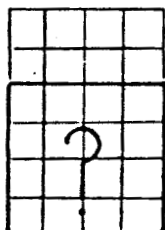
avant



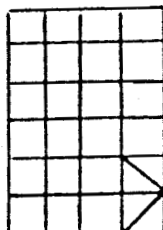
après



c) Exemples de symboles internationaux.

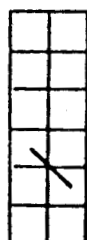


question

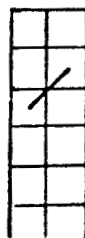


en avant

d) Arbitraires.



un
une



le
la

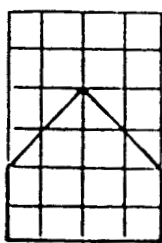
Un dictionnaire a été édité en 1980. Il comprend environ 1400 symboles, qui sont classés, soit par thèmes, soit selon les éléments de base qui le constituent.

Vingt - six classes d'éléments de base sont définis et à chacune de ces classes est attribuée une lettre de l'alphabet.

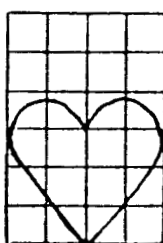
Dans ce dictionnaire les symboles sont représentés sur des grilles divisées en sous - bases.

Les symboles BLISS peuvent former des assemblages des éléments de base.

- éléments juxtaposés



maison

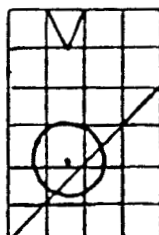


aimer

La maison que l'on aime

chez soi

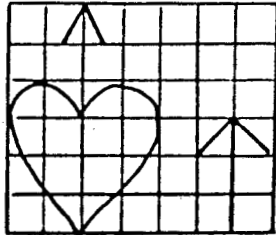
- éléments superposés



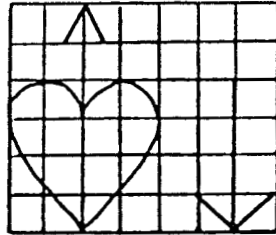
oeil barré = aveugle.



- Le sens des flèches a aussi une signification différente.

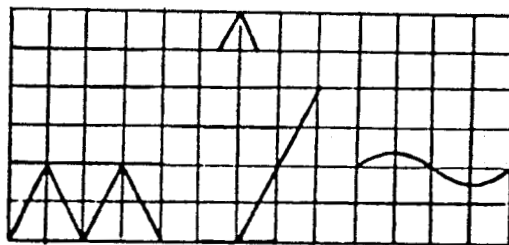


Heureux



Triste

- Exemple de phrase possible avec le BLISS.



animal

a besoin

eau

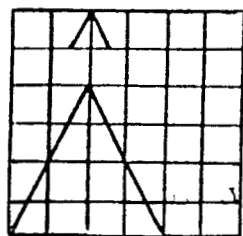


(l') animal

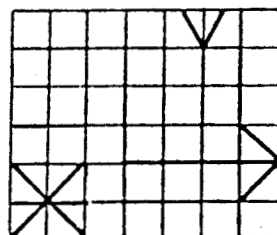
a besoin

(d') eau

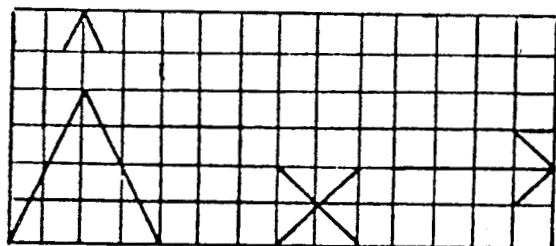
- La distance entre les symboles modifie également le sens



marche

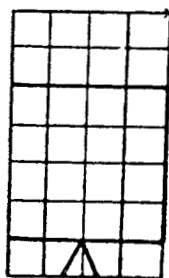


vite

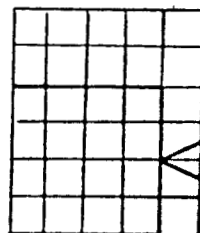


court

- La position du localisateur intervient également.



plancher



mur

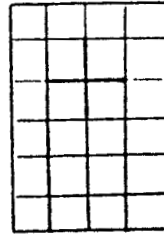
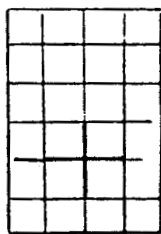
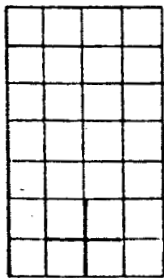


- Il existe aussi des indicateurs spéciaux, les principaux sont

∧	: indicateur d'action (verbe)	∧	: forme affirmative
)	: verbe au passé	□ ?	: forme interrogative
(: verbe au futur	- ^	: forme négative
V	: adjectif	!	: forme impérative
?	: conditionnel	<	: forme passive
×	: pluriel		
□	: chose		



- La position par rapport aux lignes de terre, de milieu et de ciel modifie le sens des symboles.



ligne de ciel

ligne de milieu

ligne de terre

appartient

et











avec

A.2. Code Pictographique du GRACH [DES. 86]

Le code du GRACH est un moyen pictographique utilisé dans l'aide à la communication pour handicapés. Il a été mis au point par le G.R.A.C.H. (Groupement de Recherche pour l'Autonomie et la Communication des Handicapés).

















Ce code est constitué de pictogrammes représentatifs des substantifs, des verbes, des adjectifs et des pronoms.


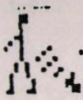









Nous présentons ci-dessous quelques pictogrammes du GRACH. La colonne au milieu est celle des pictogrammes. Le symbole BLISS correspondant au chacun des pictogrammes se trouve dans la première colonne. On peut associer chacun des pictogrammes à une phrase (ou un message) qui est dans la troisième colonne.

+!!		<i>oui, je suis d'accord</i>
?□		<i>qu'est ce que je dois faire?</i>
		<i>je voudrais aller aux toilettes</i>
		<i>j'ai soif</i>
		<i>je voudrais me laver</i>
-!!		<i>non</i>
1		<i>je voudrais voir quelqu'un</i>

		<i>j'ai faim</i>
		<i>j'ai froid</i>
		<i>j'ai trop chaud</i>
-X		<i>j'en ai assez</i>
		<i>je suis fatigué</i>
		<i>attends</i>
		<i>j'ai fini</i>



		<i>regarde</i>
		<i>je voudrais monter</i>
		<i>je voudrais sortir</i>
		<i>je voudrais descendre</i>
		<i>je voudrais rester ici</i>
		<i>je ne sais pas</i>
		<i>je voudrais mes lunettes</i>
		<i>il me manque quelque chose</i>

		<i>j'ai perdu quelque chose</i>
		<i>je ne suis pas bien installé</i>
		<i>je ne me sens pas bien</i>
		<i>je voudrais changer d'activité</i>
		<i>je voudrais jouer</i>
		<i>je voudrais un livre</i>