



**HAL**  
open science

## **EVA environnement videotex d'applications**

Patrick Bosc, Alain Chauffaut, Mario Dragone, Bertrand Hardy

► **To cite this version:**

Patrick Bosc, Alain Chauffaut, Mario Dragone, Bertrand Hardy. EVA environnement videotex d'applications. [Rapport de recherche] RT-0078, INRIA. 1987. inria-00071333

**HAL Id: inria-00071333**

**<https://inria.hal.science/inria-00071333>**

Submitted on 23 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**IRIA**

CENTRE DE RENNES

IRISQ

Institut National  
de Recherche  
en Informatique  
et en Automatique

Domaine de Voluceau  
Rocquencourt  
B.P. 105  
78153 Le Chesnay Cedex  
France  
Tél. (1) 39 63 55 11

Rapports Techniques

N° 78

**EVA**  
**ENVIRONNEMENT VIDEOTEX**  
**D'APPLICATIONS**

Patrick BOSC  
Alain CHAUFFAUT  
Mario DRAGONE  
Bertrand HARDY

Février 1987

Décembre 1986

**EVA***Environnement Videotex d'Applications**Videotex environment for applications*

Document D1 :

Publication Interne n°327

*Présentation générale*

56 pages

*Overview*

BOSC Patrick \*  
CHAUFFAUT Alain \*  
DRAGONE Mario \*\*  
HARDY Bertrand \*

- \* Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA)
- \*\* Centre Interuniversitaire de Calcul de Bretagne (CICB)

IRISA-CICB  
Campus de beaulieu  
35042 RENNES Cedex

**Résumé :** Ce document présente une méthode d'accès application-utilisateur vidéotex. Cette méthode permet de concevoir et de développer une application vidéotex indépendamment des caractéristiques du terminal d'accès. Elle autorise une évolution aisée de la présentation des dialogues et une utilisation optimale des services de visualisation de chaque type de terminal d'accès possible. Elle associe un format de présentation à chaque entrée-sortie. Elle repose sur deux logiciels. Le premier, externe à l'application, est le logiciel de composition de formats (Document D1). Il permet au maquettiste de définir la présentation de chaque entrée-sortie. Le second, intégré à l'application, est l'interface de présentation formatée (Document D2). Il prend en charge le dialogue avec l'utilisateur.

**Abstract :** This document presents a videotex user-server acces method where the application and its dialogues presentation are compiled separately. this method allows to design and to implément a videotex application idependantly of the acces terminal characteristics. It allows an easy evolutivity of the dialogues presentation and the best use of the visualization possibilities of each kind of acces terminal. It associates a presentation format to each input-output. It relies on two softwares. The first one, out of the application , is the software for format specifications (LCF Document 1). It allows the model maker to define the presentation of each input-output. The second one, integrated to the application, is the formatted presentation interface (IPF Document 2). It is in charge of the dialogue with the user.

## PREAMBULE

Une compréhension complète de ce document peut nécessiter la lecture des documents complémentaires :

- D2 = EVA - L'Interface de présentation Formatée (IPF)
- D3 = EVA - Le Logiciel de Composition de Formats (LCF)



# TABLE DES MATIERES

1	PRESENTATION GENERALE . . . . .	1-1
1.1	Introduction . . . . .	1-1
1.2	Position du problème . . . . .	1-2
1.3	Le contexte de réalisation . . . . .	1-3
1.4	Un Environnement Videotex d'Application ( EVA ) . . . . .	1-4
1.4.1	Les principes de base . . . . .	1-4
1.4.2	Une méthodologie de conception et de programmation . . . . .	1-4
1.4.3	Une méthode d'accès usager videotex . . . . .	1-5
2	L'INTERFACE DE PRESENTATION FORMATEE (IPF) . . . . .	2-1
2.1	Introduction . . . . .	2-1
2.2	Le terminal virtuel . . . . .	2-1
2.2.1	Intérêt du terminal virtuel . . . . .	2-1
2.2.2	Les communications application-terminal virtuel. . . . .	2-2
2.2.3	Les services usager . . . . .	2-3
2.3	LES FONCTIONNALITES DE L'IPF . . . . .	2-4
2.3.1	Détermination de la classe de visualisation du terminal d'accès . . . . .	2-4
2.3.2	Mise en oeuvre des spécifications de présentation. . . . .	2-4
2.3.3	Visualisation des blocs logiques d'édition . . . . .	2-5
2.3.4	La gestion des interventions de l'utilisateur. . . . .	2-6
2.3.5	Facilités de correction et d'annulation des réponses de l'utilisateur . . . . .	2-7
2.3.6	La gestion des guides . . . . .	2-8
2.3.7	Rafraîchissement de l'écran . . . . .	2-9
2.4	LES DIALOGUES USAGER-IPF . . . . .	2-9
2.4.1	La transparence des dialogues IPF-Usager pour l'application . . . . .	2-9
2.4.2	La gestion de la rangée zéro. . . . .	2-10
2.4.3	Les réponses à interprétation invariable. . . . .	2-10
2.4.4	Les réponses en fin d'affichage d'un bloc physique. . . . .	2-11
2.4.5	Les réponses en saisie monochamp. . . . .	2-11
2.4.6	Les réponses en saisie par grille. . . . .	2-12
2.4.7	Les réponses en visualisation d'un bloc visualisable. . . . .	2-12
2.4.8	Les réponses en visualisation d'un guide de saisie. . . . .	2-13
2.5	LES PRIMITIVES D'ACCES A L'IPF. . . . .	2-13
2.5.1	Ouverture de la méthode d'accès. . . . .	2-13
2.5.2	Fermeture de la méthode d'accès. . . . .	2-14
2.5.3	Ecriture . . . . .	2-14
2.5.4	Lecture. . . . .	2-15
2.5.5	Interrogation du terminal . . . . .	2-15

2.6	Le protocole d'échange de données application-IPF. . . . .	2-16
2.6.1	L'IPF et les données . . . . .	2-16
2.6.2	Données d'édition . . . . .	2-16
2.6.3	Données saisies . . . . .	2-17
2.6.4	Valeurs initiales . . . . .	2-17
2.6.5	Code intervention . . . . .	2-18
<b>3</b>	<b>LE LOGICIEL DE COMPOSITION DE FORMATS (LCF) . . . . .</b>	<b>3-1</b>
3.1	Introduction . . . . .	3-1
3.2	Le poste de travail . . . . .	3-1
3.3	Les services du LCF. . . . .	3-2
3.3.1	La maîtrise des classes de visualisation. . . . .	3-2
3.3.2	Mécanismes de transposition entre les classes. . . . .	3-2
3.3.3	Gestion des spécifications d'une classe. . . . .	3-3
3.3.4	Un langage vidéographique de description de format . . . . .	3-3
3.4	Les caractéristiques d'un format des classes videotex . . . . .	3-3
3.4.1	Type d'un format . . . . .	3-3
3.4.2	Taille d'un format ou d'un élément . . . . .	3-4
3.4.3	Fonction de mise en page inter-formats. . . . .	3-5
3.4.4	Fonctions de succession des blocs physiques . . . . .	3-5
3.4.5	Fonctions vidéographiques de visualisation . . . . .	3-5
3.4.6	Les sous-formats . . . . .	3-6
3.4.7	Fonctions complémentaires. . . . .	3-6
3.5	Les fonctions vidéographiques de visualisation . . . . .	3-7
3.5.1	Les composants d'un format . . . . .	3-7
3.5.2	Positionnement des champs. . . . .	3-8
3.5.3	Le champ littéral . . . . .	3-9
3.5.4	Le champ d'édition de donnée de longueur fixe . . . . .	3-9
3.5.5	Le champ d'édition de donnée de longueur variable. . . . .	3-9
3.5.6	La référence à un sous format. . . . .	3-10
3.5.7	Les champs de saisie de donnée. . . . .	3-11
3.5.8	Le détachement à la page . . . . .	3-12
3.5.9	Les extrémités du format . . . . .	3-12
3.5.10	Les formats " rangée zéro " . . . . .	3-13
3.5.11	Le format lecture monochamp . . . . .	3-13
3.5.12	Le format lecture-grille . . . . .	3-14
3.5.13	Les formats initialisation et écoute . . . . .	3-14
3.6	Le langage de manipulation de formats . . . . .	3-15
3.6.1	Création d'un format . . . . .	3-15
3.6.2	Destruction d'un format . . . . .	3-15
3.6.3	Modification d'un format . . . . .	3-15
3.6.4	Copie de format . . . . .	3-16
3.6.5	Synonymie de formats . . . . .	3-16
3.6.6	Liste des numéros de formats . . . . .	3-16
3.6.7	Affichage de formats . . . . .	3-16
3.6.8	Impression sur papier de formats . . . . .	3-17
3.6.9	Affichage de séquences de composantes visuelles . . . . .	3-17
3.6.10	Visualisation simulée . . . . .	3-17

3.7	Les requêtes de manipulation de contexte . . . . .	3-17
3.7.1	Le contexte de travail du maquettiste . . . . .	3-17
3.7.2	Affichage du contexte . . . . .	3-18
3.7.3	Choix du clavier de composition . . . . .	3-18
3.7.4	Choix du répertoire . . . . .	3-18
3.7.5	Choix de la classe de visualisation spécifiée . . . . .	3-19
3.8	Les requêtes de manipulation des classes de visualition . . . . .	3-19
3.9	Les requêtes complémentaires . . . . .	3-20
3.9.1	Guide des requêtes . . . . .	3-20
3.9.2	Sortie du LCF . . . . .	3-20
4	CONCLUSION . . . . .	4-1

# 1 PRESENTATION GENERALE

## 1.1 Introduction

Ce document présente un environnement de développement d'applications vidéotex interactives - EVA - développé par le Centre Interuniversitaire de Calcul de Bretagne (CICB) d'après un cahier des charges proposé par l'Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA).

Il est composé de quatre chapitres. Le premier chapitre présente le problème. Les deux suivants décrivent les deux logiciels constituant une méthode d'accès usager. Le dernier chapitre est la conclusion de ce travail.

Nous conseillons pour l'utilisation de cette méthode d'accès de suivre une méthodologie de conception et de programmation de l'application vidéotex en cours de construction. Nous étudions actuellement une méthodologie. Les personnes intéressées peuvent prendre contact avec l'équipe "Bases de Données et applications grand public" de l'IRISA.

Remarque : Attention, ce premier document n'est pas un manuel d'utilisation de EVA. Cette information fait partie des documents suivants.



## 1.2 Position du problème

La Direction Générale des Télécommunications distribue actuellement, en grande quantité, des terminaux vidéotex MINITEL. Ces terminaux possèdent des caractéristiques nouvelles par rapport aux terminaux informatiques classiques. Le dialogue avec le service informatique accédé est facilité par l'usage de touches fonctions. Ces touches fonctions représentent des messages standards pour le service qui les interprète à sa façon. Les possibilités de visualisation de ces terminaux sont riches mais difficiles à manipuler directement dans le langage de présentation fourni ( norme ANTIOPE ).

L'objectif, à long terme, est de concevoir et de réaliser un environnement de développement d'applications vidéotex interactives.

Un premier point est la définition d'une nouvelle méthode d'accès usager qui fera l'interface entre l'utilisateur vidéotex et l'application.

Un second point est la réalisation d'outils d'aide à la mise en oeuvre de cette méthode d'accès pour les réalisateurs de l'application.

Le dernier point est la définition d'outils d'aide à la conception et au développement de l'application.

Seuls les points 1 et 2 font l'objet du développement actuel; la notion d'*environnement* est réduite aux outils de mise en oeuvre de la méthode d'accès.

### 1.3 Le contexte de réalisation

Le projet EVA résulte d'une collaboration entre l'Institut de Recherche en Informatique et Systèmes Aléatoires ( IRISA ) et le Centre Interuniversitaire de Calcul de Bretagne ( CICB ). C'est une première valorisation d'une étude plus large sur la définition d'une méthode d'accès usager pour application interactive, menée au sein de l'équipe Base de Données de l'IRISA. Cette étude se poursuit actuellement par la conception d'un environnement convivial de production des dialogues d'une application interactive.

Cet environnement de développement est réalisé sous le système d'exploitation MULTICS du HB-68. Chaque application créée est également installée sur cet ordinateur. L'utilisateur d'une application se connecte à partir d'un terminal vidéotex et via le réseau TELETEL. La méthode d'accès autorise cependant l'accès à partir de tout type de terminaux grâce à la notion de classe de visualisation.

Le développement s'est déroulé en quatre étapes : l'étude de l'existant par le CICB, la définition du cahier des charges puis des spécifications globales de réalisation en collaboration IRISA-CICB, et enfin la réalisation par le CICB sur l'ordinateur HB 68-MULTICS.

Le cahier des charges offre une définition complète de la méthode d'accès usager vidéotex. Cependant, seule une restriction de cette définition est réalisée dans cette première version décrite par les documents D2 et D3. Les spécifications globales prennent tout de même en compte toute la méthode d'accès.

La réalisation est programmée dans le langage PL/1 restreint au langage C; c'est à dire en utilisant des phrases PL/1 facilement traduisibles en phrases C. Ceci afin s'assurer un transport aisé du logiciel sous UNIX.

## 1.4 Un Environnement Videotex d'Application ( EVA )

### 1.4.1 Les principes de base

Trois objectifs principaux ont guidé la définition de cet environnement de développement d'application videotex :

- Autoriser une conception de l'application largement indépendante des caractéristiques du terminal d'accès de l'utilisateur.
- Laisser des possibilités d'évolution aisée de l'application, après sa réalisation, tant au niveau des services offerts, qu'au niveau des dialogues.
- Permettre une utilisation optimum des services de visualisation de chaque type de terminal d'accès.

La conception de cet environnement repose sur deux principes :

- d'une part, séparer les problèmes liés à la programmation de l'application de ceux liés à la visualisation des dialogues.
- d'autre part, baser la structure de programmation de l'application sur des automates d'états finis.

Dans la phase de conception et de réalisation de l'application, deux acteurs interviennent : le programmeur et le maquettiste. Un troisième acteur intervient au cours d'une session d'utilisation de l'application : l'utilisateur.

L'environnement EVA propose deux outils de développement et un outil d'exploitation.

### 1.4.2 Une méthodologie de conception et de programmation

Cette méthodologie n'est, actuellement, supportée par aucun logiciel. Ceci fait l'objet d'études complémentaires. Nous proposons une méthode adaptée à la gestion de dialogues d'applications interactives et à l'usage de notre méthode d'accès.

Cette méthodologie n'entre pas dans la définition de EVA. Son usage n'est pas obligatoire mais simplement conseillé.

La conception d'une telle application est fortement dépendante des dialogues envisagés entre l'utilisateur et cette application.

La conception des dialogues s'appuie sur la construction d'un graphe de dialogues.

Le principal objectif de ce graphe est de matérialiser, de façon agréable et pratique, les informations échangées entre l'utilisateur et l'application.

Ce graphe est composé de noeuds et d'arcs orientés.

Un noeud modélise une question posée à l'utilisateur.

Un arc modélise la compréhension par le système de la réponse de l'utilisateur à la question de départ, ainsi que le traitement et les résultats associés à cette intervention. Il aboutit sur une nouvelle question.

La structure de programmation de l'application s'appuie entièrement sur le graphe des dialogues. La "navigation" de l'utilisateur dans ce graphe va être programmée à l'aide d'un automate d'états finis.

Nous proposons des règles de traduction du graphe en un automate. En complément de cette traduction, le programmeur écrira deux ensembles de procédures; l'un réalisant les actions du graphe, l'autre les interprétations des réponses de l'utilisateur pour chaque étape de dialogue.

La méthode préconisée s'appuie sur les services de la méthode d'accès usager.

Le choix du langage de programmation et le contenu des procédures sont laissés à l'initiative du programmeur. Il devra cependant respecter quelques règles d'interfaçage avec la méthode proposée, concernant notamment la gestion des événements, les communications avec l'utilisateur.

Cette première étape de conception et de réalisation débouche donc sur l'existence d'une application vidéotex qui dialoguera avec un usager au travers d'une méthode d'accès vidéotex.

### 1.4.3 Une méthode d'accès usager vidéotex

Cette méthode d'accès est le premier résultat d'une étude plus large sur la conception d'un environnement de présentation des dialogues d'une application interactive.

La méthode EVA confie à un spécialiste tous les choix de présentation visuelle des dialogues de l'application et autorise une gestion externe à l'application des choix de services demandés à la méthode d'accès. L'application et la présentation font ainsi l'objet de compilations séparées.

Le programmeur introduit dans les ordres d'entrée-sortie de l'application les données, sans aucune commande de présentation, accompagnées d'un numéro de format. Chaque ordre d'entrée-sortie possède donc un numéro de format.

Le maquettiste spécifie la présentation des ordres d'entrée-sortie à l'aide d'un logiciel de construction de formats.

Chaque format est repéré par son numéro. Un format contient toutes les commandes de présentation de l'ordre d'entrée-sortie qui fait référence à ce format, ainsi que les commandes de choix de services.

L'ensemble des formats constitue une table de présentation.

Afin de répondre à l'objectif d'une utilisation optimum des services de visualisation du terminal d'accès de l'utilisateur, nous avons classé les différents types de terminaux en fonction de leurs possibilités de visualisation. Chaque groupe constitue une classe de visualisation. Pour chaque classe, le maquettiste devra construire une table de présentation.

Les ordres d'entrée-sortie de l'application sont des appels à l'interface de présentation formatée. Cette interface réalise les services demandés par le maquettiste et notamment la mise en oeuvre de la présentation spécifiée par le maquettiste.

En cours de session, cette interface présente chaque message de l'application selon les commandes trouvées dans le format référencé dans l'ordre d'entrée-sortie associé. Ce format est pris dans la table de présentation de la classe de visualisation à laquelle appartient le terminal d'accès de l'utilisateur.

#### 1.4.3.1 L'interface de présentation formatée.

L'application dialogue donc avec l'utilisateur au travers de l'interface de présentation formaté ( IPF ). Cette interface lui rend transparent toutes les caractéristiques réelles du terminal d'accès ainsi que la présentation des informations à mettre en oeuvre sur ce terminal.

Pour l'application, le terminal de l'utilisateur se comporte de façon standard. Ce terminal virtuel accepte des messages constitués d'une liste de données. Il retourne des messages accompagnés d'un code fonction.

L'IPF fournit plusieurs services à l'application en fonction des demandes introduites par le maquettiste :

- Il détermine la classe de visualisation du terminal d'accès.
- Il édite les messages de l'application selon les commandes de présentation trouvées dans le format référencé.
- Il prend en charge la saisie de données demandée par l'application. Il peut saisir une ou plusieurs données à la fois.
- Il met en oeuvre des dialogues entre l'IPF et l'utilisateur afin de réaliser certains de ses services. Ces dialogues sont transparents à l'application.
- Il gère le découpage logique-physique. Un message de l'application est un bloc logique qui va se transformer, après la fonction de présentation, en un ou plusieurs blocs physiques, c'est à dire en une ou plusieurs pages vidéotex. L'IPF se charge de la visualisation successive des pages à l'utilisateur. Il autorise une visualisation en avant, mais aussi en arrière de ces blocs.

- Il peut gérer une partie des guides qu'il intègre avec la gestion des guides des dialogues IPF.
- Il gère la répétition de l'affichage du bloc physique courant lorsque l'utilisateur le demande, suite à une dégradation visuelle de ce bloc.
- Il prend en charge la traduction automatique du langage de présentation utilisé pour la classe de visualisation dans le langage propre du terminal d'accès.

#### 1.4.3.2 Le logiciel de composition de formats.

Le logiciel de composition de formats ( LCF ) est l'outil proposé au maquettiste pour spécifier la présentation des dialogues dans chacune des classes de visualisation, et pour choisir les services demandés à l'IPF.

L'accès à ce logiciel se fait à l'aide d'un clavier de composition de pages vidéotex.

Le LCF gère les tables de présentation des classes de visualisation considérées, il offre certains mécanismes de transposition des spécifications réalisées pour une classe en spécifications pour une autre classe, enfin il fournit un langage de manipulation et de description de formats.

L'originalité de ce poste de travail réside dans son langage vidéographique de description de format, qui permet au maquettiste de voir, immédiatement en construisant son format, le rendu obtenu à l'utilisation de ce format en cours de session de l'application. Pour cela, chaque format a une composante visuelle où le maquettiste construit une image vidéotex de ce format. Il introduit, à l'aide des fonctions du clavier de composition, sur chaque champ de donnée les attributs de visualisation qu'il veut lui affecter.

#### 1.4.3.3 La coordination Programmeur-Maquettiste.

Dans le cadre du projet EVA, aucune prise en charge informatique de cette coordination n'est prévue bien qu'elle soit nécessaire. Les incohérences relevées par l'IPF, en cours de session, ne seront pas corrigées mais interrompent la session.

Le programmeur fournit une liste des formats avec les caractéristiques des données à éditer ou à saisir.

Il informe le maquettiste de l'édition d'ensemble, c'est à dire des différentes séquences de formats correspondant aux différents scénarios possibles. Actuellement cette phase nécessite une collaboration étroite entre le programmeur et le maquettiste pour la mise au point finale des spécifications.

## 2 L'INTERFACE DE PRESENTATION FORMATEE (IPF)

### 2.1 Introduction

L'IPF assure l'interface entre l'utilisateur et l'application au cours d'une session d'utilisation de cette application. Il fournit des services tant à l'application qu'à l'utilisateur.

L'IPF répond aux objectifs suivants :

- Il assure une conception de l'application indépendante du terminal, puisque ce terminal usager apparaît à l'application comme un terminal virtuel disposant toujours de caractéristiques standards,
- Il assure une conception de l'application indépendante de la présentation des dialogues car c'est lui qui met en oeuvre, pour chaque entrée-sortie, la présentation choisie par le maquetiste,

Dans l'adaptation " terminal virtuel - terminal réel ", l'IPF prend en charge des services de visualisation et de saisie de données qui impliquent notamment des dialogues avec l'utilisateur. Ceux-ci restent, dans la mesure du possible, transparents à l'application.

Dans la suite, nous présentons la notion de terminal virtuel et l'ensemble des services assurés par l'IPF. Nous décrivons ensuite les primitives utilisables par le programmeur, le protocole d'échange de données entre l'application et l'IPF puis les dialogues entre l'utilisateur et l'IPF.

### 2.2 Le terminal virtuel

#### 2.2.1 Intérêt du terminal virtuel

Le principal terminal d'accès envisagé lors de la conception de la méthode d'accès EVA a été le terminal vidéotex MINITEL. Ce terminal présente deux caractéristiques intéressantes : de riches possibilités de visualisation de caractères et la présence de touches fonctions.

Cependant, nous ne voulions pas nous limiter à ce terminal car souvent les entreprises désirant offrir de nouveaux services télématiques accessibles à partir de MINITEL possèdent également un parc de terminaux informatiques classiques. Il est donc raisonnable de vouloir accéder à ces nouveaux services à partir des anciens terminaux.

Il faut aussi noter que ces terminaux informatiques possèdent souvent des possibilités de visualisation et des touches fonctions inexploitées des applications actuelles.

La prise en compte des touches fonctions par l'application est un point intéressant qui permet d'optimiser les dialogues par contre la prise en charge par l'application de la présentation des données en fonction de chaque terminal d'accès nous semble inutile à l'interactivité entre l'utilisateur et l'application. D'où la définition d'un terminal virtuel avec touches fonction mais sans présentation.

Il ne faut pas oublier qu'une présentation des dialogues est quant même mise en oeuvre sur chaque terminal d'accès, mais sa spécification est confiée à un spécialiste -le maquettiste-. Ce dernier dispose d'un outil qui lui permet de définir une présentation soignée pour chaque classe de visualisation. Dans une classe de visualisation, nous regroupons tous les types de terminaux ayant les mêmes possibilités de visualisation (voir chapitre 3).

## **2.2.2 Les communications application - terminal virtuel.**

Nous présentons dans cette sous-section le terminal virtuel tel qu'il est vu par le programmeur ainsi que la communication entre l'application et ce terminal virtuel.

Nous appelons message un ensemble de données échangées entre l'application et le terminal virtuel.

Nous distinguons deux types de messages :

- Le message d'édition qui est toujours accompagné d'un numéro de format. Les données contenues dans ce message seront présentées à l'utilisateur selon les directives trouvées dans le format référencé.
- le message de saisie qui est toujours accompagné d'un code fonction. Les données contenues dans ce message ont été introduite par l'utilisateur. Le code fonction indique la touche fonction utilisée par l'utilisateur pour valider la saisie de ces données.

La structure des données de ces messages respecte un protocole d'échange décrit en section 2.6.

L'application dialogue avec l'IPF à l'aide d'ordres d'Entrée-Sortie. Chaque ordre d'E/S est bloquant pour l'application.

L'ordre d'écriture transmet à l'interface un message d'édition avec son numéro de format. Lorsque les données ont été présentées à l'utilisateur le contrôle est rendu à l'application.

La présentation des données à l'utilisateur peut être longue et nécessiter la visualisation de plusieurs écrans, aussi le terminal virtuel gère un mécanisme permettant à l'utilisateur d'intervenir en cours de présentation (cf 2.3.4) Certaines de ces interventions peuvent intéresser l'application, notamment l'annulation simple de l'édition en cours ou l'expression d'une requête prévue par l'application, aussi existe-t-il un ordre de



communication qui permet à l'application de connaître la nature de l'intervention si elle a eu lieu.

L'ordre d'écoute permet donc à l'application de savoir si l'utilisateur est intervenu au cours de la présentation du dernier ordre d'écriture. Un message de saisie est transmis à l'application avec son code fonction. Eventuellement le message est vide de données si l'utilisateur n'en a pas introduit ou si il n'est pas intervenu.

L'ordre de lecture implique deux messages d'édition et un message de saisie. Un ordre de lecture permet d'acquérir une ou plusieurs données. Le premier message d'édition sert à la présentation de la question à l'utilisateur, notamment avec la position des champs de saisie. Des valeurs initiales sont éventuellement introduites dans les champs de saisie à partir du deuxième message d'édition. Les deux messages d'édition font référence au même format. Aucune possibilité d'intervention n'est laissée à l'utilisateur pendant cette phase de présentation. Lorsque les données à saisir ont été introduites et validées par l'utilisateur, le contrôle est rendu à l'application qui récupère ces données dans le message de saisie.

### 2.2.3 Les services usager

Cette sous-section liste les services pris en charge par l'IPF. Une description plus complète est faite à la section suivante. Ces derniers sont transparents à l'application, cependant le programmeur doit en connaître l'existence afin de ne pas chercher à les reproduire de façon redondante.

Certains services sont globaux à la session, d'autres particuliers à une phase de la session. Le choix des services est précisé dans les formats associés aux ordres d'E/S. En particulier, à l'ordre d'ouverture de l'IPF est associé le format initialisation (cf 3.5.13), et à l'ordre écoute est associé un format d'écoute (cf 3.5.13), qui ne contiennent pour l'instant que des choix de services.

Les informations présentées à l'utilisateur sont affichées sur l'écran de son terminal. Afin que l'utilisateur puisse lire et consulter ces informations à son rythme, ce flux d'information est contrôlé. Nous appelons bloc physique d'édition un ensemble d'informations remplissant entièrement le contenu d'un écran de l'utilisateur (cf 2.3.3). En terminologie vidéotex ce bloc physique est appelé page. Un bloc physique ne succède à un autre qu'après la confirmation de lecture du premier par l'utilisateur.

Les services concernent essentiellement la consultation des blocs physiques et l'aide à la saisie des données :

- ré-affichage du bloc physique courant lorsque celui-ci a été mal transmis sur les lignes téléphoniques,
- possibilité de re-visualiser les précédents blocs physiques de l'édition en cours,
- gestion du dialogue de saisie,

- affichage du guide associé à un champ de saisie,
- prise en compte d'une intervention en cours d'édition.

La mise en oeuvre de ces services nécessitent des dialogues entre l'utilisateur et l'IPF qui prennent en compte l'usage de certaines touches fonctions. La transparence de ces dialogues pour l'application n'est pas toujours assurée (cf 2.4), dans ce cas certaines touches fonction ne sont, momentanément, plus disponibles pour les dialogues de l'application.

## **2.3 LES FONCTIONNALITES DE L'IPF**

### **2.3.1 Détermination de la classe de visualisation du terminal d'accès**

Cette fonction s'accomplit durant la phase d'initialisation de l'IPF. Elle a pour objectifs :

- de reconnaître le terminal d'accès et en déduire sa classe de visualisation.
- de préparer l'accès à la table de présentation définie par le maquettiste pour cette classe de visualisation.

La reconnaissance du terminal d'accès et de sa classe de visualisation permet notamment à l'IPF de prendre en charge la traduction automatique du langage de présentation utilisé pour la classe de visualisation dans le langage propre du terminal d'accès. Actuellement cette fonction n'est pas mise en oeuvre pour les classes de visualisation considérées car on suppose que le langage de présentation de chaque terminal est identique à celui de sa classe.

### **2.3.2 Mise en oeuvre des spécifications de présentation.**

Une des fonctions de base de l'IPF est la présentation d'un message d'édition sur le terminal de l'utilisateur.

Cette présentation est réalisée à l'aide des commandes de présentation trouvées dans le format associé au message d'édition.

Chaque format a été construit par le LCF à partir des spécifications du maquettiste (cf chapitre 3.).

Un format est composé d'une suite de commandes que l'IPF interprétera dans l'ordre séquentiel.

Ces commandes définissent :

- les champs d'information (littéraux, données à éditer, données à saisir)
- la mise en page (saut de ligne, nouvelle page, marge, etc...)
- les fonctions globales (cloche, effacement de la rangée zéro, etc...)
- les fonctions de services (gestion de guide, mode de validation de lecture, ...)

La phase d'initialisation de l'interface met en place les services globaux demandés par le maquettiste dans le format initialisation.

La présentation comporte deux phases imbriquées : le formatage et les dialogues usager.

Le formatage consiste à construire à partir de certaines commandes du format et des données du message d'édition, une ou plusieurs chaînes de caractères codées dans le langage de présentation du terminal d'accès. de la classe de visualisation à laquelle est rattaché le terminal d'accès puis traduite dans le propre langage de ce terminal.

Un bloc logique d'édition est l'ensemble des chaînes codées générées pour un ordre d'E/S de l'application.

Les dialogues usager réalisent les choix de services, les fonctions de présentation globales, l'affichage du bloc logique d'édition, la saisie des données.

### 2.3.3 Visualisation des blocs logiques d'édition

Un bloc logique d'édition est donc l'ensemble des chaînes codées associées à un ordre d'E/S de l'application.

Un bloc physique d'édition est un ensemble de chaînes codées remplissant exactement un écran de l'utilisateur (encore appelé plan de visualisation - cf 3.5.2).

Un bloc physique est affiché à l'aide d'un ou de plusieurs ordres d'écriture sur le terminal.

L'IPF gère des blocs physiques et un bloc visualisable.

Un bloc visualisable est un ensemble de blocs physiques que l'utilisateur pourra revisualiser sur requête à l'IPF.

#### 2.3.3.1 La transformation logique-physique

Un bloc logique, en général, est soit plus grand, soit plus petit qu'un bloc physique. Lorsqu'il est plus grand, l'IPF le découpe en un ensemble de blocs physiques. Lorsqu'il est plus petit, plusieurs blocs logiques peuvent être concaténés pour constituer un bloc physique.

L'affichage d'un bloc physique ne peut commencer que si le précédent bloc physique a été validé afin de laisser le temps à l'utilisateur de consulter ce bloc.

On distingue deux modes de validation de lecture :

- par temporisation : le bloc physique est affiché pendant un certain temps puis remplacé par le bloc suivant. Cette durée d'affichage est fixée par le maquettiste.
- par l'utilisateur : l'utilisateur intervient explicitement pour valider le bloc physique. Si le bloc physique contient un champ de saisie, c'est en remplissant ce champ qu'il valide le bloc, sinon c'est en appuyant sur la touche fonction appropriée (traditionnellement SUITE).

### 2.3.3.2 La gestion du bloc visualisable.

L'IPF mémorise un certain nombre de blocs physiques présentés à l'utilisateur, pour constituer le bloc visualisable. Ces blocs forment une liste ordonnée selon l'ordre de présentation à l'utilisateur. L'utilisateur pourra consulter à sa guise le bloc visualisable courant en dialoguant directement avec l'IPF (cf 2.4).

Les délimiteurs d'un bloc visualisable sont définis à priori par des règles de l'IPF sur les ordres d'E/S de l'application, cependant le maquettiste peut inhiber certains délimiteurs en le précisant dans les formats correspondants (cf 3.4.7).

Les délimiteurs par défaut sont :

- Le début d'une session de l'application,
- Le début de la présentation d'un ordre de lecture,
- La validation par l'utilisateur d'une saisie de données demandées par un ordre de lecture,
- Une intervention explicite de l'utilisateur au cours de la présentation d'ordres d'écriture.

Par ces règles, un bloc visualisable correspond soit à un ordre de lecture, soit à l'ensemble de ordres d'écriture transmis entre deux ordres de lecture ou bien entre un ordre de lecture et une écoute de l'utilisateur.

### 2.3.4 La gestion des interventions de l'utilisateur.

Comme nous l'avons indiqué lors de la définition du terminal virtuel (cf 2.2.2), l'utilisateur peut intervenir au cours de la présentation de données d'écritures.

L'utilisateur peut demander :

- soit uniquement l'annulation de la présentation en cours,
- soit accompagner cette annulation, de données pour l'application. Dans ce cas, il faut qu'il y ait été autorisé par l'application. Cette autorisation est précisée par le maquettiste dans le format correspondant à cette présentation (cf 3.4.4).

L'IPF travaille de deux façons selon le contexte d'autorisation :

- si l'utilisateur n'est pas autorisé : un message l'en avertit et l'édition des données lui est imposée. Tout cela est transparent à l'application.
- si l'utilisateur est autorisé alors l'IPF :
  - abandonne la présentation en cours,
  - mémorise l'intervention ainsi que les données si il y a lieu,
  - rend le contrôle à l'application.

Ensuite l'IPF inhibe tous les ordres d'écriture qui lui sont transmis par l'application jusqu'à la réception d'un ordre d'écoute venant de l'application.

L'ordre d'écoute informe l'application du comportement de l'utilisateur. C'est le code fonction du message de saisie associé qui lui signifie cette intervention. Trois cas se présentent :

- si l'utilisateur n'est pas intervenu, il aura la valeur particulière "NON-INTERVENTION",
- si il a demandé juste l'annulation, il aura la valeur "ANNULATION",
- si il a fournit des données, il aura la valeur "VALIDATION".

### **2.3.5 Facilités de correction et d'annulation des réponses de l'utilisateur**

Les dialogues entre l'utilisateur et l'IPF impliquent des réponses de l'utilisateur aux sollicitations de l'IPF (cf 2.4).

Une réponse est composée d'une donnée et d'un code fonction. Le code fonction correspond à la touche fonction utilisée pour valider la réponse. Chaque donnée est constituée d'une chaîne de caractères introduite par l'utilisateur à l'aide de son clavier.

En cours de frappe, l'IPF permet d'effacer le dernier caractère de la chaîne ou d'effacer toute la chaîne. ( Ces fonctions sont généralement mise en oeuvre par les touches CORRECTION et ANNULATION sur un minitel.)

Notons le cas particulier des valeurs initiales éditées sur un champs de saisie; elles sont considérées comme des pré-réponses offertes à l'utilisateur. L'utilisateur peut alors les modifier par les fonctionnalités de correction et d'annulation et les compléter avant de les valider.

Remarque : Rappelons que, dans le cadre du réseau TELETEL, le VIDEOPAD traite la fonctionnalité de correction des caractères introduits. Cependant lorsque le vidéopad ne dispose plus de caractères à effacer, il transmet une réponse vide de donnée avec le code fonction CORRECTION. Dans un contexte de saisie, ce type de requête est utilisée pour modifier les valeurs initiales.

### 2.3.6 La gestion des guides

Les dialogues entre l'application et l'utilisateur sont liés au possibilités offertes par l'application. Celles ci ne sont pas toujours explicitement décrites par l'application dans la question posée à l'utilisateur. Par ailleurs certains services justifient des explications sur leur fonctionnement, certaines données ou certaines réponses étant, par exemple, présentées de façon codées. Tout cela explique pourquoi l'utilisateur peut avoir besoin d'une aide, d'un guide à certaines étapes de dialogues.

Nous distinguons deux types de guide :

- le mode d'emploi : qui précise à l'utilisateur toutes ses possibilités de réponses à cette étape de dialogue.
- le guide d'information ou de saisie : qui fournit plus de détails sur la nature d'un service ou d'une donnée.

Nous avons déjà signalé que les dialogues Application-Usager pouvaient être complétés par des dialogues IPF-Usager. La fonctionnalité Guide sera donc gérée à la fois par l'application et l'IPF.(cf 2.4.8).

Les modes d'emploi concernant uniquement l'IPF seront décrits en 2.4.

L'application gèrent les guides soit directement, soit en faisant appel à un service de l'IPF. Ces choix sont précisés par le maquettiste dans le format de l'E/S concernée.

Le service offert par l'IPF concerne le guide associé à un champ de saisie. Dans ce cas l'application ne s'en préoccupe plus : le maquettiste le définit, puis l'IPF l'affichera sur requête de l'utilisateur (touche fonction GUIDE sur MINITEL).

Si le maquettiste le demande, le champ de saisie associé peut être transféré en rangée zéro pendant l'affichage du guide. L'utilisateur peut alors directement répondre à la question en cours de consultation du guide.

### 2.3.7 Rafraîchissement de l'écran

Compte tenu de l'existence d'une liaison téléphonique entre le terminal d'accès et l'IPF, des parasites peuvent perturber l'affichage de l'écran. En conséquence l'IPF assure le ré-affichage du bloc physique courant à la demande de l'utilisateur.

## 2.4 LES DIALOGUES USAGER - IPF

Dans le dialogue Application-Usager, l'utilisateur envoie des requêtes à l'application, soit sur sa propre initiative au cours d'une intervention, soit en réponse à une question de l'application.

Cependant, l'IPF constitue l'intermédiaire entre l'application et l'utilisateur. C'est par lui que passent toutes les informations échangées. C'est lui qui fournit à l'utilisateur les moyens de s'exprimer. Les requêtes de l'utilisateur sont d'abord des réponses aux sollicitations de l'IPF.

Les sollicitations de l'IPF sont des questions de deux origines différentes :

- ou bien c'est une question posée par l'application que l'IPF a directement transmise à l'utilisateur,
- ou bien c'est une question nécessaire à la poursuite du service IPF en cours, notamment la validation d'un bloc physique issue de la présentation des messages d'édition de l'application.

De façon générale, la réponse de l'utilisateur se compose d'une chaîne de caractères et d'un code fonction correspondant à la touche fonction qui a servi à valider la réponse.

Dans les deux cas de figure, la réponse est pré-analysée par l'IPF, qui soit la transmet à l'application, soit l'interprète pour poursuivre le service en cours.

Dans la suite, on s'intéresse à la description des réponses interprétées et utilisées par l'IPF pour la gestion des services usager.

### 2.4.1 La transparence des dialogues IPF - Usager pour l'application

Cette sous-section pose le problème des interférences entre les dialogues Application-Usager et les dialogues IPF-Usager.

Dans les deux cas, les dialogues sont basés sur l'utilisation d'un nombre limité de touches fonction ( 8 pour le MINITEL ) combinées avec des chaînes de caractères. Aussi, si aucune précaution n'est prise, une même réponse pourrait être simultanément valable soit pour l'application, soit pour l'IPF.

Garantir la non-interférence de façon transparente à l'application, implique des dialogues plus lourds pour l'utilisateur. Aussi nous avons adopté un compromis en réservant, en fonction du contexte de dialogue, certaines réponses à l'IPF.

#### 2.4.2 La gestion de la rangée zéro.

L'utilisation de la rangée zéro est une autre source de conflit entre l'application et l'IPF.

Le principe retenu est de donner la priorité à l'application. Lorsque l'IPF utilise la rangée zéro, elle mémorise son contenu avant utilisation, puis le restaure après.

L'application dispose de deux ordres d'E/S particuliers : écriture-rangée-zéro et lecture-rangée-zéro. L'effacement de la rangée zéro est une opération gérée dans les formats de présentation de n'importe quel ordre d'E/S.

L'IPF utilise la rangée zéro à trois fins :

- message d'avertissement à l'utilisateur à propos du service en cours,
- zone de saisie de toute intervention possible de la part de l'utilisateur en cours d'édition,
- zone de saisie du choix de l'utilisateur lors de la présentation d'un guide avec saisie gérée par l'IPF.

#### 2.4.3 Les réponses à interprétation invariable.

Ces réponses sont indépendantes du contexte de dialogue ( c'est à dire quelle que soit la question posée par l'IPF ) et ne font pas avancer le dialogue ni avec l'application ni avec l'IPF.

Si l'on considère le MINITEL, il y en a deux :

- chaîne terminée par REPETITION qui provoque le réaffichage du bloc physique en cours d'édition. Si la chaîne n'est pas vide, elle est interprétée comme une réponse au champ de saisie courant. Cette chaîne est alors intégrée dans ce champs avant le réaffichage du bloc.
- chaîne terminée par ANNULATION qui signifie que cette chaîne ne doit pas être considérée comme une réponse. L'IPF efface cette chaîne sur l'écran de l'utilisateur.

Dans les deux cas, l'IPF se remet en attente d'une réponse à la question courante qui a été posée à l'utilisateur.



#### 2.4.4 Les réponses en fin d'affichage d'un bloc physique.

Rappelons que, vue de l'application, l'édition s'effectue à un niveau logique, le programmeur n'ayant pas à se préoccuper de l'encombrement de la présentation finale qui apparaîtra sur le terminal de l'utilisateur. Il y a nécessité de cadencer la succession de l'affichage des blocs physiques. L'un des modes de succession consiste à faire intervenir explicitement l'utilisateur.

L'application lui pose la question suivante " Que voulez vous faire sur cette validation de bloc physique? ". L'utilisateur dispose alors des possibilités de réponses suivantes :

- demander l'affichage du bloc physique suivant (SUITE),
- demander l'arrêt de l'affichage du bloc logique courant (ANNULATION),
- entrer dans un dialogue permettant de consulter les pages précédentes du bloc visualisable en cours d'affichage (RETOUR) (cf 2.4.7),
- répondre par une requête appropriée correspondant à une intervention s'il y est autorisé (dans le cas contraire, un message d'erreur lui est adressé),
- demander une aide pour connaître les réponses autorisées (GUIDE).

#### 2.4.5 Les réponses en saisie monochamp.

L'application pose une question à l'utilisateur dont la réponse constitue un seul champ de donnée. L'IPF présente cette question à l'utilisateur et analyse sa réponse.

Il faut noter que la présentation de la question peut s'étendre sur plusieurs blocs physiques. Dans ce cas particulier, l'utilisateur ne peut pas intervenir en fin d'affichage d'un bloc physique ni annuler la présentation.

Lors de la saisie de la donnée, les trois réponses suivantes sont interprétées par l'IPF et utilisées par l'IPF sans être transmises à l'application :

GUIDE	qui fait entrer l'utilisateur dans un dialogue de visualisation d'un guide attaché à ce champ. Si ce guide n'est pas géré par l'IPF, cette réponse est transmise à l'application, sinon la suite du dialogue est décrite en sous-section 2.4.8.
CORRECTION	si le champ de saisie comporte une valeur initiale. Le dernier caractère de cette valeur est alors effacé,
ANNULATION	si le champ de saisie comporte une valeur initiale. La valeur est totalement effacée.

#### 2.4.6 Les réponses en saisie par grille.

La gestion d'une grille de saisie est complètement transparente à l'application qui demande une lecture multichamps et récupère l'ensemble des valeurs correspondant à chacun des champs.

L'IPF présente la grille à l'utilisateur, éventuellement avec des valeurs initiales, puis le positionne sur le premier champ. La réponse de l'utilisateur lui permettra soit de déplacer dans la grille, soit d'entrer des données, soit de valider les données de la grille. Tant qu'il n'a pas validé les données, les mêmes possibilités lui sont offertes sur le nouveau champ où il s'est déplacé.

Les réponses suivantes sont interprétées par l'IPF :

- chaîne terminée par SUITE fait passer au champ suivant (on effectue une gestion circulaire des champs). Dans les trois premiers types de réponse, la chaîne vient compléter la donnée du champ courant. Si la chaîne est vide la donnée du champ est inchangée.
- chaîne terminée par RETOUR fait passer au champ précédent,
- chaîne terminée par ENVOI valide la grille globalement et provoque le transfert à l'application,
- CORRECTION efface le dernier caractère de la donnée du champ courant si elle existe (valeur initiale ou réponse précédente) sinon provoque un message d'erreur,
- ANNULATION provoque l'effacement de la totalité de la donnée du champ courant s'il en existe une (valeur initiale ou réponse précédente) sinon un message est affiché,
- GUIDE est traité de la même façon que dans le cas d'une saisie monochamp,
- toute autre réponse est refusée et provoque l'affichage d'un message d'erreur.

#### 2.4.7 Les réponses en visualisation d'un bloc visualisable.

Le bloc visualisable peut comprendre plusieurs blocs physiques. La "navigation" de l'utilisateur dans ce bloc visualisable est gérée par l'IPF de manière transparente à l'application.

Pendant cette navigation, l'IPF interprète les réponses suivantes :

RETOUR provoque l'affichage du bloc physique qui a précédé dans le temps le bloc physique déjà visualisé. Lorsque l'utilisateur atteint le début du bloc visualisable, un message d'avertissement est affiché.

SUITE                                   provoque l'affichage du bloc physique qui a suivi dans le temps le bloc physique déjà visualisé. Lorsque l'utilisateur dépasse la fin du bloc visualisable, il quitte la phase de consultation du bloc visualisable.

SUITE                                   permet de quitter directement la phase de consultation d'un bloc visualisable.

## **2.4.8 Les réponses en visualisation d'un guide de saisie.**

C'est le maquettiste qui précise qui doit gérer le guide associé à un champ de saisie de donnée - l'application ou l'IPF ?

Si la tâche est confiée à l'IPF, trois modes de gestion sont disponibles :

- mode guide inutile : L'utilisateur sera averti en rangée zéro qu'aucune information de guide n'est associée à ce champ de saisie.
- mode guide sans saisie : L'IPF affiche à l'utilisateur un ensemble de pages d'informations qui ont été construites par le maquettiste. Cet ensemble est considéré comme un bloc logique d'édition d'une part, et comme un bloc visualisable d'autre part. L'affichage se déroule avec un dialogue tel que celui défini en sous-section 2.4.4, cependant l'utilisateur ne peut pas intervenir pour s'adresser à l'application. En tant que bloc visualisable, il n'annule pas le précédent mais le suspend. Ce dernier est restauré en fin de consultation des informations du guide. Le bloc guide peut être revisualisé grâce au dialogue avec l'IPF défini en sous-section 2.4.7.
- mode guide avec saisie : C'est un mode identique au précédent, cependant l'utilisateur peut aussi répondre à la question de l'application correspondant à ce champ de saisie, à la fin de l'affichage d'un bloc physique. Une telle intervention termine la consultation du guide et valide la réponse à ce champ de saisie.

Ainsi au cours du dialogue de saisie d'une donnée, si l'utilisateur répond GUIDE, selon l'indication trouvée dans le format, cette réponse est soit transmise à l'application, soit traitée par l'IPF qui fournit le service guide choisi par le maquettiste.

## **2.5 LES PRIMITIVES D'ACCES A L'IPF.**

### **2.5.1 Ouverture de la méthode d'accès.**

Nom : OUVRIER (répertoire, nom, date, heure, statut)

Fonctionnalités :

- fournit le nom de l'utilisateur, la date et l'heure,
- détermine et initialise la table de présentation de l'application en fonction de la classe de visualisation impliquée par le type de terminal d'accès de l'utilisateur,
- localise la table de présentation d'après le répertoire donné,
- mémorise les modes de fonctionnement initiaux du terminal,
- initialise le terminal aux modes de fonctionnement définis par le maquettiste,
- retourne un statut d'ouverture relatif à la présence de la table de présentation.

**Description :**

```
proc OUVRIR fix = (car [1.N] : répertoire)
             mod = (car [1.22] : nom,
                   car [1.6] : date,
                   car [1.6] : heure,
                   ent       : statut)
```

### 2.5.2 Fermeture de la méthode d'accès.

**Nom :** FERMER (date, heure)

**Fonctionnalités :**

- fournit la date et l'heure,
- libère la table de présentation,
- restaure les modes de fonctionnement initiaux du terminal .

**Description :**

```
proc FERMER fix = ()
             mod = (car [1.6] : date,
                   car [1.6] : heure)
```

### 2.5.3 Ecriture

**Nom :** ECRIRE (nofmt, données)

**Fonctionnalités :**

- affiche les données à écrire selon les spécifications du format de numéro nofmt,
- L'utilisateur peut intervenir en cours d'édition du bloc logique, dans ce cas celle-ci est mémorisée et les ordres d'écriture suivants sont inhibés jusqu'à la récupération de l'intervention par l'application.

**Description :**

```
proc ECRIRE fix = (ent      : nofmt,
                  car [1.N] : données)
                  mod = ()
```

**2.5.4 Lecture.**

**Nom :** LIRE (nofmt, données\_à\_écrire, données\_à\_lire, valeurs\_initiales, code\_intervention)

**Fonctionnalités :**

- initialise un nouveau bloc visualisable sauf indication contraire dans le format,
- édite les données à écrire selon les spécifications du format de numéro nofmt (aucune intervention de l'utilisateur n'est possible pendant cette phase)
- édite les éventuelles valeurs initiales des données à lire
- lit les données à saisir et positionne le code intervention

**Description :**

```
proc LIRE fix = (ent      : nofmt,
                 car [1.N] : données_à_écrire,
                 car [1.M] : valeurs_initiales)
                 mod = (car [1.M] : données_à_lire,
                       ent      : code_intervention)
```

**2.5.5 Interrogation du terminal**

**Nom :** ECOUTE (nofmt, message, code\_intervention)

**Fonctionnalité :**

- transmet à l'application le comportement de l'utilisateur au cours de l'édition du bloc logique précédent,

- En cas d'intervention explicite de l'utilisateur, le bloc visualisable courant sera terminé, sauf indication contraire dans le format.

Il faut noter que l'intervention ANNULATION est également transmise à l'application si l'utilisateur l'a émise.

#### Description :

```
proc ECOUTE fix = (ent      : nofmt)
                  mod = (car [1.N] : message,
                        ent      : code_intervention)
```

## 2.6 Le protocole d'échange de données application – IPF.

### 2.6.1 L'IPF et les données

L'IPF assure deux services : le formatage des données qu'il reçoit de l'application et la saisie des données qu'il doit envoyer à l'application (cf 2.2.2).

Le formatage concerne soit de données d'édition classique, soit des valeurs initiales de données à saisir.

### 2.6.2 Données d'édition

Un message d'édition contenant des données d'édition est une chaîne de caractères dont la structure est déterminée à partir du format associé.

Le formatage de données d'édition procède de la manière suivante : L'IPF dispose de la chaîne de caractères d'une part, et du format d'autre part. Le format est composé d'une liste de directive de présentation. L'IPF analyse le format et dès qu'il a besoin d'une donnée, il extrait la sous chaîne de caractères correspondante dans le message d'édition. La correspondance s'effectue pas à pas en progressant dans le format d'une part et dans la chaîne d'autre part.

Les sous-chaînes de données sont de deux types :

- donnée de longueur fixe : c'est une suite de caractères ASCII dont le nombre est fixé dans la directive de présentation.
- donnée de longueur variable : la sous-chaîne se compose d'une suite de caractères ASCII précédée de sa longueur codée sur 4 caractères.

### 2.6.3 Données saisies

Les données saisies par l'IPF sont transmises à l'application sous la forme d'une chaîne de caractères structurée de la façon suivante : C'est une liste de sous chaîne représentant chacune une donnée saisie. L'ordre des données correspond à l'ordre de positionnement des champs de saisie de haut en bas et de gauche à droite.

Chaque sous chaîne comporte deux zones :

- la longueur de la donnée saisie codée sur 4 caractères,
- les caractères composant la donnée saisie.

La zone longueur doit être interprété par l'application avant usage. 4 cas sont définis selon la valeur de cette zone :

0	La valeur du champ a été effacée,
9999	la valeur initiale de ce champ n'a pas été modifiée, en particulier ce champ est resté vide si la valeur initiale était nulle,
9998	L'utilisateur demande des informations de guide sur ce champ, sans en avoir modifié le contenu,
0<lg<7999	C'est la longueur de la donnée explicitement introduite,
7999<lg<9998	L'utilisateur demande des informations de guide sur ce champ, et une donnée de longueur lg-8000 a été introduite.

Dans les trois premiers cas, la deuxième zone de la sous chaîne est supprimée.

### 2.6.4 Valeurs initiales

Les valeurs initiales sont fournies par l'application à l'IPF. Ce dernier les présente dans les champs de saisie selon les directives du format associé.

Ce message d'édition a une structure similaire à un message de saisie. C'est une liste de sous chaîne de caractères représentant chacune une valeur initiale. L'ordre de correspondance est celui du positionnement des champs de saisie.

Chaque sous chaîne est composée de deux zones :

- la longueur de la valeur initiale. Si le programmeur n'a pas de valeur initiale pour le champ correspondant, cette valeur est nulle.

- la chaîne de caractères de la valeur initiale.

Si le programmeur n'a aucune valeur initiale à afficher, il lui suffit de donner une longueur nulle pour ce message d'édition dans son ordre d'E/S.

### 2.6.5 Code intervention

Il est retourné par l'IPF à la suite d'une demande de lecture ou d'écoute, ce code signifie la touche fonction utilisée par l'utilisateur pour valider sa réponse ou son intervention.



# 3 LE LOGICIEL DE COMPOSITION DE FORMATS (LCF)

## 3.1 Introduction

Le LCF permet au maquettiste de spécifier la présentation des dialogues de l'application concernée et de préciser les services demandés à l'IPF.

Ce chapitre décrit le poste de travail du maquettiste (section 3.2), puis les services offerts par le LCF (section 3.3). La section suivante fournit les caractéristiques d'un format (section 3.4). Ensuite c'est une présentation du langage vidéographique de description de format (section 3.5), suivi de celle du langage de manipulation de formats (section 3.6). Les dernières sections présentent les requêtes de manipulation de contexte (section 3.7), les requêtes de manipulation de classes (section 3.8), et des requêtes complémentaires (section 3.9).

## 3.2 Le poste de travail

Le maquettiste travaille à partir d'un clavier de composition.

Ce poste est constitué du clavier, d'un écran de télévision, de lecteur de disquettes, d'une imprimante couleur. Ce poste est en liaison informatique classique avec le LCF installé sous MULTICS.

Ce poste dispose de deux modes de travail :

**le mode calculateur :** Ce mode permet de dialoguer avec le calculateur par un échange de messages constitués d'une chaîne de caractères ASCII.

**le mode composition :** Ce mode est local au poste, c'est à dire qu'il n'implique aucun échange avec le calculateur. Ce mode permet de construire des pages vidéotex. Une page vidéotex est une image remplissant un écran de visualisation. Elle est composée de caractères visualisables affectés d'attributs de visualisation. Un caractère visualisable est une forme qui est affichée sur l'écran. Un attribut de visualisation est une fonction de présentation affectée à un caractère visualisable - par exemple la taille, la couleur, le clignotement, etc... -. On distingue les attributs parallèles qui peuvent varier à chaque nouveau caractère, et les attributs séries qui sont affectés à tout une zone contigue de caractères. Cette zone débute par un espace appelé délimiteur. Pour plus de détails, on se reportera aux " spécifications vidéotex de visualisation et de codage -Edition DAI 1980, CCETT RENNES ".

Dans la suite du document, le mot caractère signifiera à la fois la forme et les attributs de visualisations affectés à cette forme.

Une fonction de normalisation des pages vidéotex au standard LCF autorise le travail à partir de tout clavier de composition.

Dans le même but, une fonction d'adaptation du dialogue est installée, notamment pour autoriser l'usage du MINITEL comme clavier de composition en collaboration avec le logiciel de composition de pages vidéotex MECANOTEX.

### **3.3 Les services du LCF.**

#### **3.3.1 La maîtrise des classes de visualisation.**

Une classe de visualisation regroupe tous les terminaux ayant les mêmes possibilités de visualisation d'informations, indépendamment des différences dans les langages de présentation utilisés sur chacun de ces terminaux.

Le LCF connaît trois classes de visualisation :

- vidéotex couleur,
- vidéotex noir et blanc,
- visu sans présentation.

Une spécification de visualisation sera définie pour chacune des classes de visualisation autorisées par l'application.

Actuellement tous les terminaux informatiques classiques sont regroupés dans la classe "visu sans présentation". Cependant, une classe de visualisation "visu avec présentation" est envisagée dans le LCF. Elle disposera des attributs de teinte, demi-teinte, inverse-vidéo, cligotement, lignage.

#### **3.3.2 Mécanismes de transposition entre les classes.**

Afin de simplifier et d'accélérer la spécification dans plusieurs classes, le LCF offre deux services de transposition automatique des spécifications réalisées pour une classe dans une autre classe.

- Duplication des spécifications de la classe "vidéotex couleur" dans la classe "vidéotex noir et blanc", et réciproquement.
- Traduction d'une des classes vidéotex dans la classe "visu sans présentation" par suppression de tous les attributs vidéotex. C'est actuellement la seule façon de spécifier la visualisation associée à cette classe.

La cohérence entre les spécifications est à la charge du maquettiste.

### **3.3.3 Gestion des spécifications d'une classe.**

Le maquettiste doit construire tous les formats introduits par le programmeur dans l'application.

L'interface entre le programmeur et le maquettiste n'est pas prise en charge par l'informatique. C'est une opération manuelle : le programmeur fournit une liste des numéros de format avec pour chaque format le type des données et leur sémantique. Une coopération entre le programmeur et le maquettiste est nécessaire au départ pour expliquer les services offerts par l'application, et à la fin pour la mise au point d'ensemble de la présentation.

Le LCF offre deux langages de spécification : un langage de manipulation des formats et un langage de description d'un format.

Le LCF construit et gère les tables de présentation qui seront utilisées par l'IPF. Il fournit les moyens d'une liaison automatisée entre ces tables et l'IPF.

### **3.3.4 Un langage vidéographique de description de format**

Le langage de description d'un format a deux aspects. L'un classique qui consiste à affecter des fonctionnalités de présentation globalement au format ou de choisir des services, l'autre visuel - c'est l'originalité de ce langage - qui définit la présentation des champs d'information.

Dans cette composante visuelle, le maquettiste construit une image vidéotex de son format qui sera interprétée par le LCF puis traduite dans une représentation interne directement utilisable par l'IPF. L'avantage de cette méthode est que le maquettiste voit immédiatement en construisant son format, le rendu obtenu lors d'une session d'utilisation de l'application. Cette fonctionnalité est justifiée par l'apparition de terminaux ayant des possibilités de visualisation de plus en plus riches. Le maquettiste s'exprime directement " en couleur ". Ce découpage fonctionnel Programmeur-Maquettiste fait apparaître le maquettiste comme un spécialiste de la présentation.

## **3.4 Les caractéristiques d'un format des classes videotex**

### **3.4.1 Type d'un format**

Il y a 7 types de formats :

édition-rangée-zéro : C'est un format d'édition dans la rangée zéro.

édition-écran : C'est un format d'édition dans le reste de l'écran, de la rangée 1 à la rangée 24.

lecture-monochamp : C'est un format de saisie. Il y a éventuellement une édition de données puis une lecture de caractères dans un seul champ de saisie.

lecture-rangée-zéro : C'est un format de saisie monochamp dans la rangée zéro.

initialisation : C'est un format de service. Il est unique et sert à préciser les services globaux à la session à mettre en oeuvre dans l'IPF.

écoute : C'est un format de service relatif aux interventions de l'utilisateur.

lecture-grille : C'est un format de saisie ayant plusieurs champs de saisie. Etant donné les services actuels du réseau TELETEL, la gestion des grilles de saisie n'offrira pas toutes les garanties de contrôle de non-débordement des champs habituellement fournies.

Des éléments complémentaires sont parfois rattachés à un format. Il y a deux types d'éléments complémentaires :

l'élément de remplacement : C'est une description d'édition utilisée automatiquement par l'IPF dans la mise en oeuvre d'un format. Plusieurs éléments de remplacements sont regroupés en un sous format.

l'élément guide : C'est une description d'édition utilisé sur une réponse guide de l'utilisateur. Cet élément est associé à une saisie d'information.

### 3.4.2 Taille d'un format ou d'un élément

La taille d'un format ou d'un élément se définit à partir de sa composante visuelle, en fonction du nombre de rangées ou de pages vidéotex nécessaires à sa description.

La taille est liée au type du format ou de l'élément.

Les formats édition-rangée-zéro, lecture-rangée-zéro, lecture-grille, et l'élément de remplacement se décrivent sur une seule page.

Les formats édition-écran, lecture-monochamp, et l'élément guide se décrivent, éventuellement, sur plusieurs pages.

Les formats d'initialisation et d'écoute n'ont, actuellement, pas de composante visuelle.

### 3.4.3 Fonction de mise en page inter-formats.

Ces fonctions expriment l'agencement et la mise en page des messages formatés les uns par rapport aux autres au cours d'une session d'utilisation de l'application.

Il y a trois fonctions :

- juxtaposition : Le message formaté commence sur la position courante du curseur laissée par le précédent message formaté.
- nouvelle ligne : Le message formaté commence en première colonne de la rangée suivante de celle de la position du curseur laissée par le message précédent.
- nouvelle page : Un effacement d'écran est effectué et le message formaté commence en première colonne de la première rangée.

### 3.4.4 Fonctions de succession des blocs physiques

Un bloc logique d'édition pouvant se découper en plusieurs blocs physiques, le maquettiste précise le mode de validation du bloc courant et le mode d'affichage du bloc suivant.

Les modes de validation :

- opérateur : L'utilisateur de l'application doit valider le bloc courant par la frappe de la touche SUITE. Toute autre réponse est interprétée comme il se doit, soit par l'IPF, soit par l'application.
- temporisation : Le bloc courant est affiché pendant un certain temps fixé par le maquettiste, puis automatiquement le bloc suivant est affiché. Ce mode implique que l'utilisateur ne peut plus intervenir, en particulier les possibilités d'anticipation sont exclues.

Les modes d'affichage :

défilement vers le haut,

- effacement : Après validation, l'écran est effacé et l'affichage reprend en première colonne de la première rangée.

### 3.4.5 Fonctions vidéographiques de visualisation

Ces fonctions servent à construire l'image de chaque page vidéotex constitutive d'un format ou d'un élément. Elles font l'objet de la section 3.5 .

### 3.4.6 Les sous-formats

Une des fonctions vidéographiques permet de signaler la référence à un complément de description, externe à la page en cours de construction. Ce complément est appelé sous-format.

Un sous-format est constitué de plusieurs pages vidéotex, chacune construite à l'aide des fonctions vidéographiques avec quelques restrictions. Chaque page permet d'introduire un élément de remplacement.

En cours de session, une donnée particulière du message de l'ordre d'E/S faisant appel à un sous-format est associée à ce sous-format. En fonction de la valeur de cette donnée, un des éléments de remplacement sera utilisé pour poursuivre le formatage du message. Après épuisement de cet élément de remplacement, l'IPF reprend les spécifications du format pour continuer le formatage.

Il y a deux modes de correspondance "valeur-élément de remplacement" :

par valeur : A l'élément de remplacement est associé une chaîne de caractères de longueur fixée par le maquettiste. Lorsque cette chaîne est reconnue dans le message par l'IPF, cet élément est utilisé.

par condition : A l'élément de remplacement est associée une condition que doit remplir la valeur de la donnée associée trouvée dans le message pour que cet élément soit utilisé. Une seule condition est actuellement disponible "valeur numérique supérieure à 1".

Etant donné nos modes de correspondance, une valeur peut impliquer deux éléments de remplacement différents, aussi les éléments de remplacement sont ordonnés. L'IPF respectera cet ordre dans l'examen des associations; dès qu'il trouve une association, il s'arrête. L'ordre est établi par le maquettiste.

La donnée associée dans le message à une longueur fixe en nombre de caractères. Cette longueur est précisée par le maquettiste.

Un format peut faire référence à plusieurs sous-formats.

### 3.4.7 Fonctions complémentaires.

Les fonctions de présentation globales à un format :

- cloche : L'affichage du message formaté à l'aide d'un format utilisant cette fonction commence par un signal sonore.
- effacement de la rangée zéro : Nous proposons trois fonctions d'effacement. Leur utilisation est fonction du format d'hébergement.

- effacement avant affichage : La rangée zéro est effacée avant l'affichage du message ayant référencé le format d'hébergement. ( utilisation : écran, monochamp, grille ).
- effacement après saisie : Dans une opération de saisie, soit dans la rangée zéro, soit dans l'écran, l'effacement de la rangée zéro se fait après que l'utilisateur ait validé sa réponse. ( utilisation : lecture-rangée-zéro, monochamp, grille ).
- effacement automatique : Après l'utilisation d'un format en rangée zéro, l'effacement peut se faire automatiquement par l'IPF, après un délai fixé par le maquettiste. ( utilisation : édition-rangée-zéro, lecture-rangée-zéro ).

Les fonctions de choix de service :

- inhibition de frontière de bloc visualisable : En supprimant des délimiteurs, le maquettiste peut choisir la portée des blocs visualisables.(cf 2.3.3.2).
- gestion des guides : Le maquettiste choisit parmi un des quatre modes définis, - application, inutile, automatique sans saisie, automatique avec saisie -. (cf 2.4.8)
- mode de présentation : Deux modes sont possibles.
  - + mode formattage : C'est le service principal décrit dans ce document.
  - + mode mise au point : L'IPF réalise alors une présentation sans formattage. Il édite les données brutes envoyées par l'application, sans utiliser le format référencé. Chaque ordre d'E/S entraîne l'affichage du numéro du format et des chaînes de caractères du message.

### 3.5 Les fonctions vidéographiques de visualisation

#### 3.5.1 Les composants d'un format

Un format est composé de fonctions globales ( composante affectée ), d'une ou plusieurs pages de spécifications ( composante visuelle ) et d'un ou plusieurs éléments complémentaires (composante complémentaire ).

Une page de spécification est constituée d'un certain nombre de champs d'information.

Les champs sont contigus, et se succèdent de gauche à droite et de haut en bas.

Il y a quatre types de champs : le champ de littéral, les champs d'édition, les champs de saisie, les champs de référence à un sous-format.

### 3.5.2 Positionnement des champs.

Dans le cadre des applications vidéotex, le support de visualisation pour l'utilisateur comme pour le maquettiste est un écran; c'est à dire un plan limité par quatre cotés- le haut, le bas, le gauche , le droit.

Appelons plan de spécification l'écran du maquettiste et plan de visualisation l'écran de l'utilisateur en cours de session.

Lorsqu'il définit une page de format, le maquettiste va positionner ses champs dans le plan de spécification. Quelle sera alors la position de ces champs exploités dans le plan de visualisation? Ils peuvent subir des translations horizontales et verticales.

#### 3.5.2.1 Position dans le plan de spécification

Le maquettiste choisit ses éléments de référence en verticale et en horizontale :

positionnement fixe : la référence en verticale est une rangée du plan, la référence en horizontale est le coté gauche du plan. C'est le positionnement par défaut des champs.

positionnement relatif : les références se font par rapport à un emplacement de caractère. La référence en verticale est alors la rangée passant par ce caractère, et la référence en horizontale est alors la colonne passant par ce caractère. Ce type de positionnement est utilisé après un champ dont la longueur effective est inconnue à la spécification. Nous dirons qu'une zone affectée de ce type de positionnement est attachée au champ de longueur variable. Cela induit la notion de détachement où l'on revient à un positionnement fixe.

Successivement, il peut y avoir plusieurs positionnements différents dans une page de spécification ou dans l'ensemble des pages de spécifications d'un format.

#### 3.5.2.2 Transposition dans le plan de visualisation.

Cas du positionnement fixe : La rangée de référence peut subir une translation verticale vers le haut ou vers le bas. Le coté gauche reste superposé au coté gauche du plan de visualisation. Tous les champs restent stables par rapport à cette rangée et au coté. Si une partie du plan de spécification déborde dans le cas d'une translation vers le bas, elle fera alors partie du prochain bloc physique.

Cas du positionnement relatif : La rangée de référence peut subir une translation verticale vers le haut ou vers le bas. On a la même règle de débordement que pour la transposition d'un positionnement fixe. La référence horizontale peut subir une translation horizontale vers la gauche ou vers la droite. Toute la zone attachée à ce caractère subira le même déplacement. Si l'attachement se poursuit sur plusieurs rangées, toutes ces rangées subissent le déplacement avec report des débordement soit sur



la rangée précédente si c'est un déplacement vers la gauche soit sur la rangée suivante si c'est un déplacement vers la droite.

### 3.5.3 Le champ littéral

Le LCF utilise les 9 caractères suivants de façon particulière :  
& \ | ← → \$ ↑ @

Ces caractères sont réservés si ils sont utilisés dans un certain contexte qui sera précisé dans la suite de ce document.

Toute suite jointive de caractères non réservés constitue un champ littéral qui sera fidèlement reproduit en cours de session.

### 3.5.4 Le champ d'édition de donnée de longueur fixe

Ce champ est signalé par autant de caractères réservés & que sa longueur en nombre de caractères. Chaque caractère placé dans ce champ par l'IPF héritera des attributs de visualisation affectés au & correspondant dans le format.

### 3.5.5 Le champ d'édition de donnée de longueur variable.

Ce champ est signalé par un seul caractère & immédiatement suivi du caractère \ . La fin de la description de ce champ est marquée par un nouveau caractère \ .

En cours de session, une chaîne de caractères de longueur quelconque, précisée par l'application, viendra se placer dans ce champ à partir de la position du caractère & . Elle recevra les attributs de visualisation qui ont été affectés au & dans le format. Il n'y a aucune relation de longueur entre la taille de l'espacement entre les deux caractères \ du format et la longueur réelle de la chaîne introduite depuis le message.

L'espace entre les deux caractères \ du format est d'une taille quelconque, au gré du maquettiste.

Cet espace est, éventuellement utilisé pour spécifier des positions de marge gauche et droite, la mise en valeur de sous-chaîne dans la chaîne extraite du message.

#### 3.5.5.1 Attachement-détachement.

Plusieurs champs peuvent être attachés à une donnée de longueur variable. C'est à dire que leur positionnement dépend du dernier caractère de cette donnée qui devient l'emplacement de référence. Ce caractère est symbolisé par le deuxième caractère réservé \ dans le plan de spécification.

La fin de l'attachement est marquée par la séquence réservée `\|`. On revient donc à un positionnement fixe avec la rangée où est le caractère `|` comme référence verticale. C'est un détachement à la ligne. La partie de la rangée à droite de la séquence `\|` ne peut être utilisée pour définir un champ.

Si le maquettiste ne désire pas de d'attachement après une donnée de longueur variable, il fait suivre immédiatement le caractère `\` de la séquence `\|`.

#### 3.5.5.2 Définition des marges.

Si l'ensemble des données formatées à partir d'un champ de donnée de longueur variable et des champs qui lui sont attachés encombrant plusieurs rangées, alors à partir de la deuxième rangée peuvent être définies une marge à gauche et une marge à droite.

la position du premier caractère de la marge gauche est marquée par le caractère `←`.

la position du dernier caractère de la marge droite est marquée par le caractère `→`.

#### 3.5.5.3 Mise en valeur de sous chaîne.

Lors du formatage d'une donnée de longueur variable, des attributs de visualisations sont affectés aux caractères de la chaîne correspondante cependant l'IPF peut rechercher dans cette chaîne une ou plusieurs sous-chaînes encadrées par deux délimiteurs spécifiés et leur attribuer des attributs de visualisation différents. Les délimiteurs seront ou ne seront pas visualisés selon le choix du maquettiste.

Pour cela le maquettiste introduit trois caractères consécutifs quelque part entre les deux caractères `\` du champ de donnée de longueur variable concerné.

Viennent dans l'ordre, le caractère réservé `$` qui porte les attributs à affecter aux sous-chaînes, le caractère délimiteur gauche, le caractère délimiteur droit. La demande de visualisation des délimiteurs se fait en doublant le caractère `$`.

Les attributs possibles sont restreints aux attributs parallèles et la taille des caractères doit être identique à celle de ceux du champ.

#### 3.5.6 La référence à un sous format.

Dans toute page de spécification d'un format le maquettiste peut faire référence à un sous format. Le LCF détectera ces références à l'interprétation de cette page et engagera un dialogue avec le maquettiste pour la spécification de chacun des éléments de remplacement.

Un élément de remplacement est défini par un mode de correspondance (valeur ou condition) et une page de spécifications ne contenant que des

littéraux et éventuellement un champ de donnée de longueur fixe. Ce champ n'existe que dans le mode de correspondance par condition; l'IPF vient alors y mettre la valeur de la donnée associée.

Nous distinguons deux types de sous formats en fonction de leur encombrement dans le format principal :

- sous format d'encombrement fixe : Tous les éléments de remplacement ont la taille précisée dans la référence. Par taille nous entendons encombrement en emplacements de caractères. Ce type de sous format est signalé par un nombre de caractères réservés correspondant à la taille du sous format. Les attributs parallèles autres que ceux de la hauteur des caractères ne sont pas significatifs. Cependant si plusieurs sous formats sont jointifs, c'est par un changement d'attribut du caractère que cela est indiqué.
- sous format d'encombrement variable : Les éléments de remplacement n'ont pas tous la même taille. Ce type de sous format est signalé par le caractère † immédiatement suivi du caractère \ . La fin de la description de champ est marquée par un nouveau caractère \. Seuls les attributs série et l'attribut " hauteur de caractère " affecté au † sont significatifs dans la référence. C'est cette hauteur qui sera utilisée pour gérer les lignes introduites par le format de remplacement en cours de session. La fonctionnalité d'attachement de un ou plusieurs champs à ce sous format est utilisée de la même façon que pour les champs de donnée de longueur variable (voir 3.5.5.1).

### 3.5.7 Les champs de saisie de donnée.

Le champ de saisie délimite une zone où l'utilisateur viendra frapper des informations.

Cette zone sera éventuellement matérialisée en cours de session par des caractères d'appel. Un seul type de caractère d'appel est utilisé par une application, il est défini dans le format initialisation.

Cette zone peut contenir des valeurs initiales si le programmeur les a prévues. Elles sont cadrées à gauche et se substituent aux caractères d'appel si ils existent. Une valeur initiale doit être considérée comme une pré-réponse offerte à l'utilisateur. Elle doit être acceptée ou modifiée puis validée.

Un guide pour l'utilisateur doit être associé à chaque champ de saisie. Le LCF propose quatre modes d'association de guide au maquetiste :

- retour application : L'intervention " guide " n'est pas gérée par l'IPF mais retransmise à l'application.
- inutile : Le maquetiste considère qu'un guide pour l'utilisateur est inutile sur ce champ. Un message d'avertissement sera affiché par l'IPF en rangée zéro si l'utilisateur intervient par la touche guide.

- guide automatique sans saisie : Lorsque l'utilisateur intervient par la touche guide, si le maquettiste veut que l'application lui affiche des informations statiques, il construit pour ce champ un élément guide. Ensuite cet élément est géré directement par l'IPF qui l'affiche sur toute réponse guide de l'utilisateur.
- guide automatique avec saisie : Le principe est le même que précédemment, mais l'utilisateur peut répondre au champ de saisie en cours de consultation du guide affiché par l'IPF.

#### 3.5.7.1 Champ de saisie de longueur fixe.

Ce champ est signalé par un nombre de caractères réservés @ égal à la taille du champ. Tous ces caractères ont les mêmes attributs de visualisation. En cours de session, cette zone sera d'abord éditée avec des caractères d'appel et les valeurs initiales. Ensuite le curseur se positionnera à la fin de la valeur initiale si elle existe.

#### 3.5.7.2 Champ de saisie de longueur variable.

Ce champ est signalé par un caractère @ immédiatement suivi du caractère \. La fin de la description est marquée du caractère \. En cours de session, cette zone est d'abord éditée avec la valeur initiale si elle existe. Le curseur se positionne alors derrière. Il n'y a pas de caractère d'appel. Lorsque ce champ est utilisé dans un format c'est le dernier champ de ce format.

Remarque : Si un champ de saisie porte des attributs série, il doit débiter par un délimiteur.

### 3.5.8 Le détachement à la page

Lorsque le maquettiste spécifie uniquement des champs de longueur fixe, il contrôle parfaitement la mise en page d'un écran à l'autre. Ce n'est plus le cas lorsqu'il introduit des champs de longueur variable. Aussi lorsqu'il veut qu'une partie de la présentation soit impérativement cadrée en haut et à gauche d'un écran, il introduit dans la page de spécifications précédente un détachement à la page. Cette fonctionnalité est symbolisée par la séquence \↓ et c'est le dernier caractère visualisable de cette page. Les rangées suivantes de cette page de spécifications ne sont plus significatives. Le LCF impose alors au maquettiste de poursuivre sa spécification sur une nouvelle page.

### 3.5.9 Les extrémités du format

Un format est constitué d'une ou plusieurs pages de spécifications.

Le début du format est le caractère situé en haut et à gauche de la première page de spécifications.

La fin du format est le dernier caractère visualisable autre que le caractère espace, blanc sur fond noir, simple taille, de la dernière page de spécifications.

Notons que dans les pages intermédiaires, les 24 rangées sont systématiquement considérées, sauf si un détachement à la page a été introduit.

### **3.5.10 Les formats "rangée zéro"**

Un message édité dans la rangée zéro utilise systématiquement la technique dite du "chenillard".

Le chenillard permet d'éditer dans une fenêtre de taille fixe, un message plus long que cette fenêtre. Le message est constitué de deux parties : le train et le récapitulatif.

Le train, d'une longueur quelconque, défile une fois dans la fenêtre, de la droite vers la gauche.

Le récapitulatif qui est de la taille de la fenêtre, après le défilement du train reste affiché dans la fenêtre.

Dans le cas de la rangée zéro, la fenêtre est de 30 caractères.

Le format rangée zéro est spécifié sur une page. Le train en premier, puis le récapitulatif, séparés par au moins une ligne vierge.

Le format du train peut être vide ou bien contenir des champs de littéral, d'édition de longueur fixe, d'édition de longueur variable. Sont exclus des spécifications les marges, les références à des sous formats, les champs de saisie, et les attributs de visualisation "série".

Le format du récapitulatif peut être vide ou ne contenir que des champs de littéral. Lorsqu'il est vide, la queue du train n'est pas effacée en cours de session.

#### **3.5.10.1 le format Edition-rangée-zéro.**

C'est la mise en oeuvre du chenillard précédemment décrit.

#### **3.5.10.2 le format lecture-rangée-zéro.**

C'est également une mise en oeuvre du chenillard, mais le récapitulatif contient en plus un champ de saisie de longueur fixe.

### **3.5.11 Le format lecture monochamp**

Le format monochamp est un format de saisie d'information spécifié sur une ou plusieurs pages de spécifications.

donnée, et un seul  
de longueur variable.  
ications.

ur fixe, d'autres champs  
on, l'utilisateur déborde de ce  
message d'avertissement est  
ouvelle validation.

En cours de saisie, sur un bloc insecable, aussi elle ne  
sur fixe sauf, éventuellement le  
saisie de longueur variable.

En cours de saisie, l'utilisateur créera automatiquement un nouveau bloc  
physique. Sur un bloc insecable, la saisie de longueur variable, si l'utilisateur  
déborde du bloc, l'utilisateur aura glissement du bloc physique pour  
conserver la réponse sans perte de quelques premières lignes.

### 3.5.12 Le format lecture – grille

Le format grille est un format de saisie d'informations spécifié sur une  
page de spécifications.

Il contient des champs de littéral, d'édition de donnée de longueur  
fixe, de référence à des sous-formats d'encombrement fixe et plusieurs  
champs de saisie de donnée de longueur fixe.

En cours de session, l'appel à la saisie se fait sur le premier champ en  
partant du coin haut et gauche du plan de visualisation. La saisie est  
gérée par l'IPF selon un dialogue standard (voir 2.4.6).

### 3.5.13 Les formats initialisation et écoute

Actuellement, ces formats ne présentent pas de composante visuelle.

Le format initialisation précise :

- le mode de présentation ( formatage ou mise au point ).
- le mode d'affichage ( défilement ou effacement ).
- le caractère d'appel de saisie.

Un format écoute sert uniquement à inhiber les frontières de bloc  
visualisable.

## 3.6 Le langage de manipulation de formats

Le langage de manipulation de formats est un ensemble de requêtes permettant de construire ou de connaître les formats d'une table de présentation donnée.

Les formats sont repérés par leur numéro d'ordre dans la table de présentation.

A tous moments d'une commande, le maquettiste peut interrompre et annuler l'exécution de celle-ci.

### 3.6.1 Création d'un format

Le maquettiste introduit un nouveau format dans la table.

Il définit :

- son numéro,
- ses fonctionnalités affectées,
- ses pages de spécifications,
- ses formats complémentaires.

### 3.6.2 Destruction d'un format

`destruction no_fmt -synonyme`

Toutes les informations relatives au numéro de format précisé par le maquettiste sont détruites. Ce numéro est alors libre à la création dans cette table.

Les synonymes ne sont affectés par cette commande que si l'option `-synonyme` est présente. Ils sont alors tous détruits.

### 3.6.3 Modification d'un format

Le maquettiste peut redéfinir toutes les composantes d'un format existant.

Chaque élément du format peut être remplacé ou supprimé. De nouveaux éléments peuvent être ajoutés.

### 3.6.4 Copie de format

copie no\_fmt\_source no\_fmt\_cible

Le maquettiste définit un nouveau format ( format cible ) à partir de la copie de toutes les composantes d'un format déjà existant (format source ). Par la suite ces deux formats ont une vie complètement indépendante.

### 3.6.5 Synonymie de formats

synonyme no\_fmt\_source no\_fmt\_cible

Le maquettiste définit un nouveau format (format cible ) comme étant identique à un autre format déjà existant (format source ) et cela tout au long de leur vie. C'est à dire qu'une même spécification peut être référencée à partir de deux numéros différents.

synonyme no\_fmt\_source -liste

Le maquettiste peut obtenir une liste de tous les formats synonymes d'un format donné (format source) en remplaçant le numéro cible par l'option -liste.

### 3.6.6 Liste des numéros de formats

liste no\_fmt\_inférieur no\_fmt\_supérieur

Une liste des numéros de formats existants dans la table de présentation manipulée est affichée, dans l'ordre croissant des numéros entre deux numéros précisés par le maquettiste.

Par défaut, si il n'y a pas de numéros tous les numéros existants sont affichés. Si il y a un seul numéro, il est considéré comme la borne inférieure.

### 3.6.7 Affichage de formats

Le maquettiste demande l'affichage sur son écran de tout ou partie des composantes d'un format donné.

Une variante de cette requête est un affichage successif de plusieurs formats dans l'ordre croissant de leur numéro, à partir d'un numéro donné. Seuls les formats existants sont affichés.



### 3.6.8 Impression sur papier de formats

```
imprimer no_fmt_inférieur no_fmt_supérieur
```

Le besoin s'est fait ressentir d'avoir une trace papier d'une table de présentation mais ce service n'est pas réalisé. Il nécessite la maîtrise d'une imprimante couleur par le système MULTICS. Un palliatif consiste à utiliser une imprimante couleur en recopie d'écran du terminal de composition mais c'est très lourd à manipuler.

### 3.6.9 Affichage de séquences de composantes visuelles

```
agencer no_fmt_1 [ no_fmt_2 no_fmt_3 ... ]
```

En spécifiant un format, le maquettiste a un rendu de la présentation d'une Entrée-Sortie de l'application. Pour avoir un rendu plus global de la présentation des dialogues, ce service affiche de façon enchaînée les composantes visuelles de plusieurs formats en tenant compte des composantes affectées de ces formats. C'est le maquettiste qui précise la suite des numéros de formats concernés.

Ce service n'est pas réalisé actuellement.

### 3.6.10 Visualisation simulée

```
simuler no_fmt_1 [ no_fmt_2 no_fmt_3 ... ]
```

Le LCF simule l'usage des formats spécifiés au cours d'une session usager de l'application, soit avec des données générées par le LCF, soit des données fournies par le maquettiste.

La simulation se fait avec un ou plusieurs formats selon la demande du maquettiste.

Ce service n'est pas réalisé actuellement.

## 3.7 Les requêtes de manipulation de contexte

### 3.7.1 Le contexte de travail du maquettiste

Le contexte de travail du maquettiste est composé de 4 éléments :

- le clavier de composition,
- le répertoire d'hébergement des classes de visualisation sous multics,

- les classes de visualisations existantes dans ce répertoire,
- la classe courante en cours de manipulation.

A l'appel du LCF, un contexte par défaut est considéré. Libre au maquettiste de le garder ou de le modifier à tout moment par les requêtes adéquates.

Le contexte par défaut est donc :

- un clavier FIET EDIT 400,
- le Working\_dir comme répertoire,
- la classe " vidéotex couleur " est la classe courante si elle existe, sinon un message proposant de la créer est affiché au maquettiste.

### 3.7.2 Affichage du contexte

contexte

Cette requête affiche les 4 éléments du contexte de travail.

Cette requête est automatiquement exécutée à l'appel du LCF.

### 3.7.3 Choix du clavier de composition

clavier nom\_clavier

Le maquettiste précise le type de son clavier de travail.

clavier -liste

Cette requête affiche les noms des claviers de composition pouvant être utilisés avec le LCF.

### 3.7.4 Choix du répertoire

repertoire nom\_repertoire

Le lcf lance une commande MULTICS de changement de répertoire vers le répertoire dont le nom est fourni dans la requête du maquettiste.

### 3.7.5 Choix de la classe de visualisation spécifiée

classe nom\_classe

Le maquettiste précise la table de présentation qui sera utilisée pour la manipulation des formats, depuis cet instant jusqu'à un nouveau choix. Si il n'existe pas encore de table de présentation pour cette classe, une proposition de création est affichée au maquettiste.

classe -liste -présence

Dans cette requête d'affichage d'informations, l'option :

liste                    affiche le nom de chaque classe de visualisation connue du LCF, avec ses attributs de visualisation et la liste des types de terminaux qui lui sont rattachés.

présence                affiche le nom des classes pour lesquelles il existe une table de présentation dans le répertoire de travail.

### 3.8 Les requêtes de manipulation des classes de visualition

transposition nom\_classe\_source nom\_classe\_cible

Le mode de transposition des spécifications réalisées pour une classe afin de servir de spécifications pour une autre classe est fonction des classes concernées.

Nous distinguons actuellement deux cas :

vidéo-vidéo            Les spécifications réalisées pour une classe vidéotex peuvent servir de spécification à l'autre classe vidéotex. Les nouvelles spécifications annulent et remplacent les précédentes si elles existaient. Les deux tables poursuivent leur vie indépendamment. Dans cette opération seuls les attributs "couleurs" sont transformés en attributs "gamme de gris" ou inversement. Les autres attributs ne sont pas modifiés.

vidéo-visu             Les spécifications fournies pour une des classes vidéotex servent à construire la table de présentation de la classe "visu sans présentation". Pour cela tous les attributs de visualisation sont supprimés. Le plan de spécification reste large de 40 caractères bien que le plan de visualisation sera large de 80 caractères. Seules les données exploitées à travers des champs de longueur variable et leur zone attachée viendront s'afficher dans la zone des 40 derniers caractères du plan de visualisation.

## **3.9 Les requêtes complémentaires**

### **3.9.1 Guide des requêtes**

guide nom\_requête

Pour chaque requête d'un des langages de manipulation, sur demande du maquettiste, le LCF affiche des informations sur la syntaxe de cette requête et sur son utilisation.

### **3.9.2 Sortie du LCF**

sortie

Le LCF s'arrête. Le maquettiste revient au niveau des commandes MULTICS.

## 4 CONCLUSION

Ce document est donc une présentation générale d'une méthode d'accès pour des applications interactives grand public.

Cette méthode d'accès se veut générale, c'est à dire autorisant l'utilisation de tout type de terminal, grâce à la notion de classe de visualisation. Cependant sa réalisation est très orientée "accès vidéotex", en particulier avec un terminal français MINITEL à travers le réseau français TELETEL.

Pour l'utilisation de cette méthode dans une application, vous devrez lire deux documents complémentaires. Le premier présente les services offerts par l'interface de présentation formatée (IPF). Cette interface est un module, relié à l'application, qui prend en charge toute la présentation des entrées-sorties de l'application selon les directives d'un format associé. Le deuxième document décrit les commandes du logiciel de composition de formats que vous utiliserez pour construire les formats de votre application.

Cette méthode d'accès peut être utilisée soit pour adapter une application existante au vidéotex, soit pour développer une nouvelle application. Dans le premier cas, on peut supposer le problème de la gestion des dialogues résolu. Le travail d'adaptation consiste alors simplement à transformer les ordres d'entrées-sorties selon les règles de cette méthode d'accès, puis à construire les formats correspondant. Dans le second cas, nous conseillons d'intégrer le travail précédent avec une méthodologie de conception de dialogues et de programmation de l'application réalisant ces dialogues.

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique